AD-A084 802 MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTE--ETC F/G 9/5
CAUSAL AND TELEOLOGICAL REASONING IN CIRCUIT RECOGNITION.(U)
SEP 79 J DE KLEER N00014-75-C-0643
UNCLASSIFIED AI-TR-529 NL

1 OF 3
AD
A084802

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AI-TR-529	2. GOVT ACCESSION NO. AD-A084 802	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Causal and Teleological Reasoning In Circuit Recognition.		5. TYPE OF REPORT & PERIOD COVERED Technical Report.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Johan de Kleer		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0643, NSF-MCS77-04828
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 217
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd Arlington, Virginia 22209		12. REPORT DATE Sep[illegible]979 13. NUMBER OF PAGES 216
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Causal Reasoning
Qualitative Reasoning
Electronics
Circuit Design
Circuit Recognition
Troubleshooting
Qualitative Simulation
Teleology

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis presents a theory of human-like reasoning in the general domain of designed physical systems, and in particular, electronic circuits. One aspect of the theory, causal analysis, describes how the behavior of individual components can be combined to explain the behavior of composite systems. Another aspect of the theory, teleological analysis, describes how the notion that the system has a purpose can be used to aid this causal analysis. The theory is implemented as a computer program, which, given a

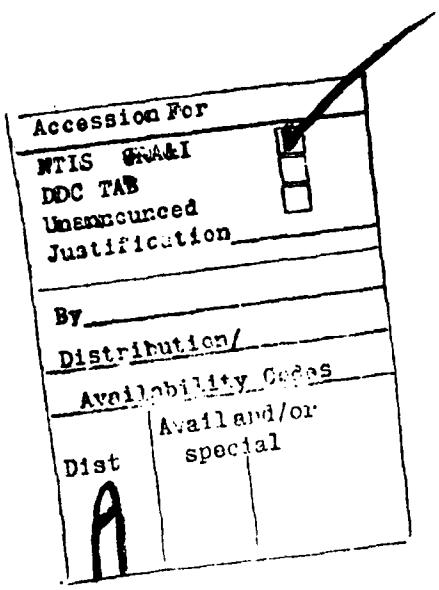
DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

circuit topology, can construct by qualitative causal analysis a mechanism graph describing the functional topology of the system. This functional topology is then parsed by a grammar for common circuit functions. Ambiguities are introduced into the analysis by the approximate qualitative nature of the analysis. For example, there are often several possible mechanisms which might describe the circuit's function. These are disambiguated by teleological analysis. The requirement that each component be assigned an appropriate purpose in the functional topology imposes a severe constraint which eliminates all of the ambiguities. Since both analyses are based on heuristics, the chosen mechanism is a rationalization of how the circuit functions, and does not guarantee that the circuit actually does function. This type of coarse understanding of circuits is useful for analysis, design and troubleshooting.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advance Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643 and in part by National Science Foundation Grant MCS77-04828.



Causal and Teleological Reasoning

In Circuit Recognition

by

Johan de Kleer

Massachusetts Institute of Technology

September 1979

This report is a revised version of a dissertation submitted to the Department of Electrical Engineering and Computer Science on January 10, 1979 in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

ABSTRACT

This thesis presents a theory of human-like reasoning in the general domain of designed physical systems, and in particular, electronic circuits. One aspect of the theory, causal analysis, describes how the behavior of individual components can be combined to explain the behavior of composite systems. Another aspect of the theory, teleological analysis, describes how the notion that the system has a purpose can be used to aid this causal analysis.

The theory is implemented as a computer program, which, given a circuit topology, can construct by qualitative causal analysis a mechanism graph describing the functional topology of the system. This functional topology is then parsed by a grammar for common circuit functions. Ambiguities are introduced into the analysis by the approximate qualitative nature of the analysis. For example, there are often several possible mechanisms which might describe the circuit's function. These are disambiguated by teleological analysis. The requirement that each component be assigned an appropriate purpose in the functional topology imposes a severe constraint which eliminates all of the ambiguities. Since both analyses are based on heuristics, the chosen mechanism is a rationalization of how the circuit functions, and does not guarantee that the circuit actually does function. This type of coarse understanding of circuits is useful for analysis, design and troubleshooting.

ACKNOWLEDGEMENTS

I would like to express my thanks to all of the people who encouraged and aided me in this work:

Gerald J. Sussman, my thesis supervisor, for providing guidance when I needed it, and for an endless series stimulating arguments;

John S. Brown, for his constant excitement about what I was doing, and for helping me articulate my ideas when I did not know what they were;

Marvin Minsky, for providing the intellectual point of view that made this type of research possible, and for being an insightful reader;

Patrick H. Winston, my second reader, for his contribution to the organization of the thesis, and for his effort to maintain the environment where this research was possible;

Seymour Papert, for first suggesting that I examine how people reason about physical systems;

Ira Goldstein, for supporting my investigation into how students reason about simple mechanics;

Ken Forbus, for reading unreadable first drafts and giving useful comments, and for believing that envisioning is important;

Jon Doyle, for reading early drafts;

Gerry Roylance, for reading early drafts, and for knowing more Electrical Engineering than I do;

Ray Reiter, for first suggesting that I study artificial intelligence, but who probably did not expect that I would write a thesis like this one;

Marilyn de Kleer, for contributing to the presentation, for putting up with my randomness, and for forcing me to remain a real person while I was working on the thesis.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 The Thesis

This thesis presents a theory of human-like reasoning in the general domain of deliberately designed physical systems. Formal quantitative theories have been developed to explain the behavior of electrical and mechanical systems, but these theories bear little resemblance to the informal qualitative reasoning of humans. For example, Network Theory is very powerful and general, but an engineer only uses it as a last resort, and then only to restricted subproblems. Most of the time he employs informal and qualitative techniques. The generality and apparent simplicity of the quantitative theories originate from their use of mathematics. People are very bad at the kind of symbol manipulation required by these theories. Instead they employ a variety of strategies to reason about engineered systems. This research develops a computational theory of two of the fundamental strategies observed in human reasoning about electrical circuits, envisioning and teleological reasoning.

Envisioning is a qualitative simulation of the system under study. The result of the envisioning is a *mechanistic* argument consisting of a sequence of events occurring in the functioning of the physical system where each event can be causally related to events earlier in the sequence. Each event is an assertion about some behavioral parameter of some constituent of the system (e.g. a change of current through a transistor). Although this apparent causality imposes a temporal order on the events, there need not be any actual time flow involved. The temporal order and the assignment of causality are entirely in the mind of the understander which need have little to do with what is actually the case. Nevertheless this rather mythical understanding is crucial in reasoning about physical systems. Elementary questions about a system's behavior can be answered directly by envisioning, and the mechanistic argument provides the foundation for more sophisticated reasoning about a system.

Since qualitative simulation describes behavior in only limited detail, it discovers multiple

mechanistic arguments for the same system. In deliberately designed systems the intended overall behavior, or *teleology*, can be used to resolve the ambiguity. The correct argument is the one which exhibits the intended behavior. Constituent objects of designed systems are grouped together in particular ways so that the purpose of the system is achieved. Since electrical circuits have been studied extensively, a fairly complete taxonomy of these groupings and their purposes has been developed. Knowledge of this taxonomy aids in resolving the ambiguities since those event sequences which cannot be accounted for by this taxonomy are probably incorrect.

Envisioning and teleological reasoning are only two aspects of the strategies humans use to reason about physical systems. However, these two are sufficient to explain a wide range of phenomena.

The central aim of this research is the development of a calculus for the causal reasoning involved in envisioning and an associated calculus for teleological reasoning. These two calculi and their interaction are explored in the context of recognition. The task of *recognition* is to determine, from a description of the structure of a system, a description of the mechanism by which the system achieves its behavior. Electrical engineering has a formal language for representing electrical systems: the circuit schematic. Since electrical engineering does not have a formal representation for mechanisms, I will develop an ontology for the representation of mechanisms that is consistent with the current engineering literature. Since the central goal of this research is to study causal and teleological reasoning and not recognition, the recognizer built on the calculi never reasons in terms of the topology or geometry of the circuit itself. Recognition thus serves as a task to evaluate the informative content of the two calculi. Although topological pattern matching, as well as geometry, certainly plays a role in human recognition of circuits, any recognizer built solely on topological pattern matching will ultimately fail. (Chapter 2 presents a detailed argument of this.)

The circuits considered for recognition are amplifiers, logic gates and regulated power supplies. The recognition focuses on the dc behavior of these circuits and ignores the ac and transient aspects; thus rf amplifiers or switching power supplies are not considered. Applications to analysis, troubleshooting and design will be evident.

Success on the recognition task is determined by whether circuits can be identified and whether the explanations for the circuits' behavior are similar to those an engineer would give. Another test of the plausibility of envisioning and teleology is how complex a recognition mechanism based

on them must be. Electrical circuits have been studied for a long time and, as a consequence, have a great deal of structure. The theory of qualitative reasoning presented here must tie directly into this structure. Current analysis, troubleshooting and design programs face difficulties which originate from their inability to understand circuit behavior at a more qualitative level. The theory of qualitative reasoning developed here should help these programs overcome their difficulties.

The remaining sections of this chapter consist a scenario, a reader's guide, a presentation of my methodology, and a discussion of related work. Chapter 2 presents an overview of a recognition process, and chapter 3 discusses the theory underlying it. Chapter 3 makes little reference to electronics, and the overview in chapter 2 can be understood without dealing with the electrical details. I suggest that the reader who knows no electronics quickly skim the remainder of chapter 1 and read chapter 3, followed by chapter 2.

1.2 Scenario

I have constructed a program QUAL based on envisioning and teleological reasoning which recognizes circuits. QUAL is written in Maclisp and can run on both the Artificial Intelligence Laboratory's PDP-10 and the Greenblatt Lisp Machine [Weinreb & Moon 79]. QUAL is completely working and has been run on hundreds of examples. Unless otherwise indicated, every part of the theory has been implemented. The program generated every example without any direct assistance from me. The following three scenarios illustrate some of its current capabilities.

The circuit schematic is presented as an unannotated topological description and is accompanied with a notation indicating the relevant input and output quantities of the circuit. If the goal of the recognition was to determine the type of the circuit (e.g. "amplifier"), this notation of the input-output quantities essentially characterizes circuit's type. However, QUAL's goal is to determine the mechanism by which that amplifier amplifies, not just that the circuit is an amplifier. Although the examples are presented in part via an English dialogue, QUAL has no natural language input-output capability. The data structures that QUAL constructs to describe circuit behavior are too complex to present in this introduction. Hence I will take the liberty of summarizing these in English (later in the thesis, the actual data structures that QUAL constructs will be presented).

The first circuit is a simple feedback amplifier. The following description is all that QUAL is told about the amplifier and the Lisp description is only included to show exactly what QUAL

is told about the circuit it is to recognize.

```
SCENARIO 1:
 (circuit: ce-feedback
  nodes: (vcc ground b1 c1 output e2 fp)
  devices: ((q1 (npn-transistor emitter: ground base: b1 collector: c1))
            (q2 (npn-transistor emitter: e2 base: c1 collector: output))
            (rc1 (resistor vcc c1))
            (rc2 (resistor vcc output))
            (rb1 (resistor e2 fp))
            (rb2 (resistor fp ground))
            (rf (resistor fp b1))
            (input (terminal b1))
            (output (terminal output))
            (common (terminal ground))
            (supply (battery vcc ground)))
  input:    (current input)
  output:   (voltage output ground))
```

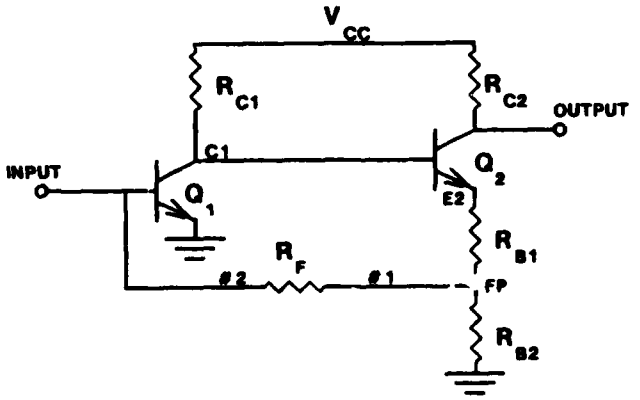


Figure 1 : Schematic for CE-FEEDBACK

I have followed standard electrical conventions for naming devices and nodes, but QUAL does not utilize these hints embedded in the symbol names. For example, RF is a Feedback

Resistor, RC1 is the Collector Resistor for Q1, VCC is the main Voltage source, RB1 is Bias Resistor 1, node C1 is tied to the Collector of Q1, and node FP is a Feedback Point. A more complete program would detect these hints as well as use geometric and topological information. Node VCC or device RB1 could have be named G0001 and G0002 without having any effect on QUAL's ability to recognize the circuit.

The dialog commences with QUAL's description of the circuit's behavior. QUAL produces its explanations in a formal language which I abridge and summarize here for readability. The scenarios are included here to be suggestive of QUAL's reasoning ability, and not of any natural language or question answering capabilities. Hence, these descriptions may seem a little terse to one not familiar with electrical circuits. When the QUAL's determination these explanations is presented in subsequent chapters, the electrical details of the dialogs will become clearer.

The circuit is a noninverting amplifier with high power gain. It has a low input impedance and high output impedance. The feedback action tends to stabilize the circuit's current gain.

What is the purpose of RC1?

RC1 is the load for Q1 which is functioning in the common-emitter configuration.

What is the purpose of RF?

RF couples the output signal to the input. Without it the circuit would not exhibit feedback.

Describe the feedback path.

The input signal is first amplified by a common-emitter stage. Q2 functions as a common-collector stage coupling the output current through RB1 and RF back to the input. This feedback samples the output current and compares this to the input current, thereby causing the amplifier to have stable current gain, higher output impedance and lower input impedance.

What contributes to the circuit's power gain?

The cascaded common-emitter stages exhibit both current and voltage gain. Therefore the circuit has power gain.

SCENARIO 2:

In order to recognize this circuit QUAL must be told that the input signal is the changing current to the load, and that the output signal is the voltage delivered to this load. QUAL then tries to identify the mechanism by which the circuit strives to have no incremental output and recognizes this mechanism to be a series-pass regulator.

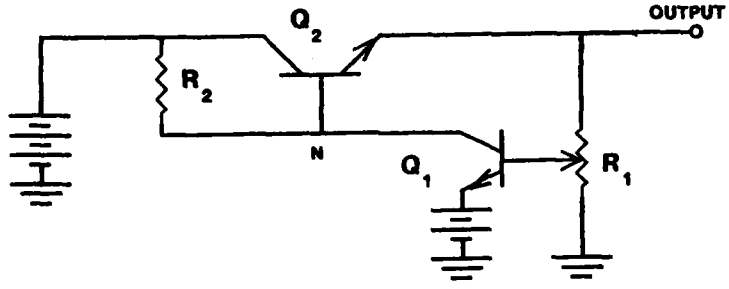


Figure 2 : Simple Power Supply

What kind of power supply is this?

A series-pass feedback regulator.

Describe how the feedback action regulates the output voltage.

Suppose the output voltage drops. This signal is coupled through R1 to Q1. Q1 compares this to the reference and begins to turn off. This increases the base drive to the series-pass element which then delivers more current to the load.

These explanations result from a combination of two very different descriptions QUAL constructs for the circuit's behavior. One description is purely causal: "The output voltage drops, causing the voltage at the base of Q1 to drop. Since the emitter of Q1 is held fixed, the base-emitter voltage of Q1 drops. This begins to turn off Q1 and lowers its collector current. As a consequence the voltage at node N rises. This causes the base-emitter voltage of Q2 to increase more than it normally would. As a consequence the output current rises." The teleological

description uses such concepts as "couple, compare, series-pass element." People do not distinguish between these two aspects of explanation and intermingle them in their explanations. Although QUAL has strategies to generate such intermingled explanations, its explanations are very crude. I constructed the text of the scenarios by starting with QUAL's crude explanation and augmenting them with other information in QUAL's data-base.

QUAL cannot generate these explanations. It knows a sufficient amount about the causality and teleology of the circuit's it recognizes to derive these explanations, but it cannot combine them into an elegant explanation. The problem of balancing how much teleology and causality to include in an argument depends on how much electronics the hearer knows. For example, if the hearer is unfamiliar with power supplies he will not understand "series-pass element." The problem of generating an appropriate explanation from a complete description of the system being explained is the subject of other research. The explanations QUAL constructs are comprehensible, but not as satisfying as possible. QUAL could far more easily be extended to a recognizer of explanations of circuit behavior than a generator of good explanations of circuit behavior.

Teleology provides a method of grouping components by purpose. This final example illustrates the use of abstraction in recognizing a complex power supply. Figure 3 is block diagram for the simple power supply of figure 2:

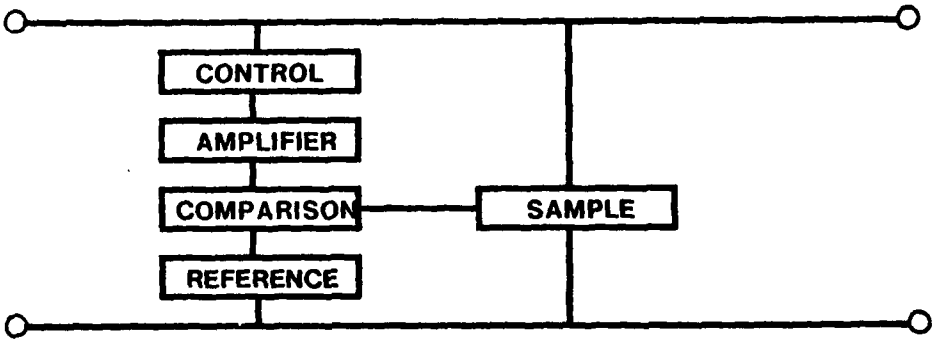


Figure 3 : Block Diagram for Power Supply

SCENARIO 3:

The following power supply has the same abstract description.

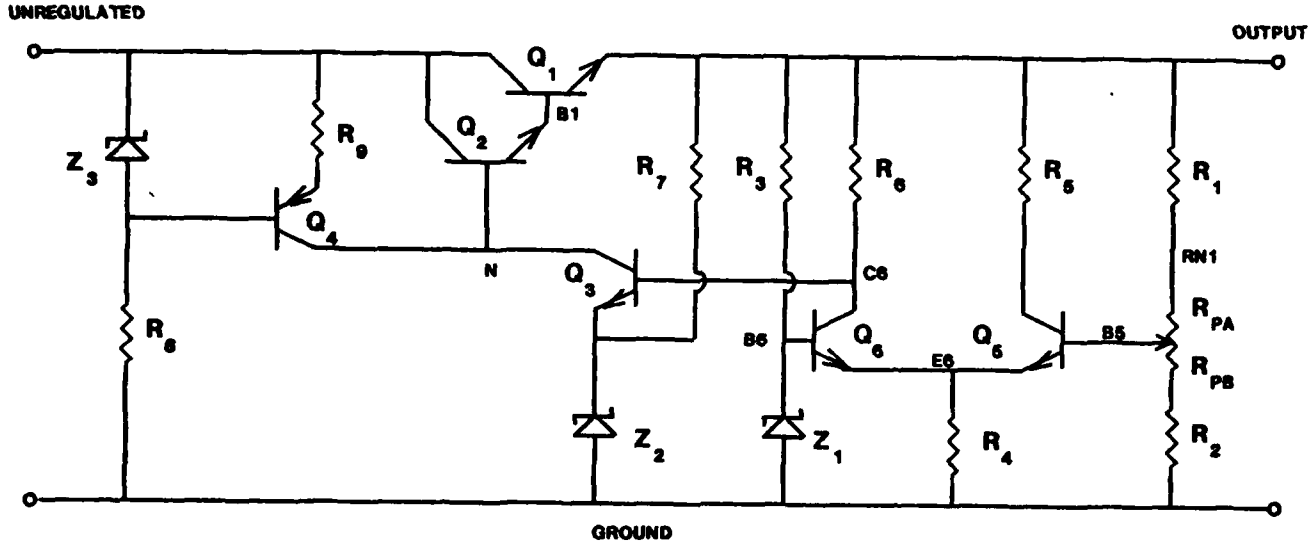


Figure 4 : Complex Power Supply

This circuit poses no problem for QUAL.

What is Z1?

D1 is part of the reference whose voltage is compared to the output voltage.

What is the purpose of Q4?

Q4 provides a constant current to the control element.

This complex circuit further illustrates the need for abstraction and teleology since the causal argument for the feedback action would fill an entire page.

1.3 Reader's Guide

The thesis is organized around a development of a calculus for causal and teleological reasoning applied to the problem of recognition of electrical circuits. Chapter 4 presents a theory of causal reasoning. The specific shortcomings of this theory when applied to recognition and

solutions to these shortcomings are explored in the remaining chapters. One style of presentation is used throughout the thesis: a simple mechanism is posited to achieve a particular goal, and then modified as specific problems appear. In general, the problems are dealt with by introducing more and more abstract descriptions of circuit behavior.

To reach the widest audience, I have, as far as is possible, employed only simple and elementary electronics. Nevertheless, the depth of the reader's appreciation of the details of the theory will be influenced by his familiarity with electronics. Those who know no electronics will find the initial sections of chapter 2 and all of chapter 3 informative and the remainder of the thesis difficult to follow. Readers with a limited understanding of electronics should have little difficulty in following the examples. Since this thesis is about how people understand circuits, these readers may in fact gain a better understanding of circuits through reading it. Sections which assume a background in electrical circuits are denoted by a *.

The following is a brief summary of the contents of the thesis:

Chapter 1 : Introduction

The objectives and methodology of the research are presented.

Chapter 2 : Overview and Background

An overview of QUAL's recognition process is presented. The framework of the ideas is given independently of electronics.

Chapter 3: Theoretical Foundations

The theory underlying the causal and teleological calculus is presented independently of electronics.

Chapter 4 : Causal Reasoning

A theory is developed of the causal reasoning exhibited by engineers.

Chapter 5 : Interpretations

The definition of a "point of view" on circuit behavior, a surprisingly subtle problem, is explored.

Chapter 6 : Feedback

The global mechanism of feedback fits directly into the theory of causal reasoning.

Chapter 7 : Teleology

Considering the purpose of a circuit helps the recognizer distinguish the intended point

of view.

Chapter 8 : Abstraction

In order to deal with more complex circuits, the behavior of the circuit must be understood at shallower levels of detail.

Chapter 9 : Conclusions

The results are summarized and the limitations of QUAL are discussed.

1.4 Methodology

The most common technique that science uses to describe physical laws is mathematics. Since classical mathematics is better suited for describing constraints than mechanisms, this has resulted in a focus on constraints on behaviors rather than on the mechanisms by which behaviors are achieved. This research attempts to describe the more qualitative and informal techniques that humans use naturally to reason about these mechanisms. The notion of a computational process forms the foundation by which these mechanisms and human reasoning about them is described.

My approach is different from the methodology of knowledge engineering [Feigenbaum 77]. Although Feigenbaum's methodology is descriptive in that he attempts to capture the reasoning of experts, he makes no attempt to characterize the reasoning humans tacitly use. Instead he constructs a stimulus-response model consistent with expert's behavior on a narrow range of problems (and therefore is very successful on that range). Since his model is nonhierarchical, it is incapable of reasoning about the same system from different perspectives and at different levels of detail. In short, his methodology is a kind of behaviorism: he does not consider the internal calculus that humans use to be important or relevant. I believe that because of this lack of concern for internal structure his methodology will fail to elucidate the true scope of human expertise. For example, all of the current systems in his methodology cannot answer simpler versions of questions in qualitatively simpler ways. My methodology is to use the computer metaphor to explore the tacit calculus that humans use. This thesis will show how reasoning from multiple points of view and at multiple levels of detail is useful and consistent with behavior observed in engineers.

The motivation for this approach comes from cognitive psychology, education and artificial intelligence. Larkin [77] observed that expert physicists use tacit and qualitative knowledge more

extensively than neophytes working on the same problems. In general, humans do not appear to utilize the formally prescribed techniques in their thinking processes. A knowledge of how humans understand would have tremendous impact in designing teaching strategies to enable students to learn more effectively [Brown *et.al.* 77]. It would therefore also play a key role in computer coaches [Goldstein 77]. The experience of artificial intelligence research has been that systems which embody even a great deal of the classical knowledge fail at tasks successfully accomplished by humans with the same knowledge. An extreme example of this is Macsyma [77]. This system can perform manipulations, usually using standard techniques, which even a mathematician would find difficult; yet it rarely does what you want it to do. (See [de Kleer & Sussman 78] for a discussion of some of the problems Macsyma has when applied to electrical problems; the major points of which are summarized in section 2.5.) The major reason for the failure of these programs can be traced to the lack of more qualitative common-sense knowledge.

Humans prefer to understand systems in terms of causes and effects rather than simultaneous constraints. Until recently no techniques have existed to describe the processes which result from expressing a behavior in terms of causes and effects. The computer metaphor provides such a technique. The computer metaphor impacts this research in three ways:

1. The human reasoning process can be viewed as an information processing system.
2. Cause-effect interactions in physical systems can be viewed as processes.
3. It is methodologically useful to construct computer programs based on 1 and 2.

This research develops a formal theory of the informal qualitative reasoning humans appear to use in understanding electronic circuits. The reason electronics was chosen over other domains is that the structure of electrical circuits is well-understood. Powerful simplifying ideas are apparent, and that the understanding of this domain is of itself useful and important. This large body of experience provides information about how people reason about circuits. These constraints will be enumerated later in this section.

Although the theory has been implemented in the program QUAL, I do not claim that the Lisp code is a theory of anything. The presentation in the following chapters leaves out most of the implementation details. The objective of this research is to identify the key concepts which underlie human reasoning in electronics independent of any formalism. I want to maximize the constraints imposed by the domain, not those imposed by the idiosyncracies of some formalism. The following chapters are organized around the ideas, not the program. One of the shortcomings

of this research is the lack of a clean distinction of where the theory stops and the implementation begins. A rough approximation of this *architectural* boundary exists at the level where QUAL ceases to record explanations for its deductions.

There are variety of different sources of information which constrain the structure of any theory. These constraints or *forcing functions* are criteria that any alleged theory must meet. Some forcing functions are behavioral in that they specify that a certain behavior has to be met. Other forcing functions suggest what the mechanisms which produce the behavior must be like. These latter structural types of forcing functions are the more interesting.

One forcing function is performance. Does it work? Any theory has to successfully recognize circuits. Although this forcing function appears to be behavioral, it turns out to have considerable impact on the structure of the theory. Since the architectural boundary is far removed from the actual Lisp code, the implementation of tentative theories is very time consuming as well as seemingly unrewarding. However, the fact that the architectural boundary is so far removed from an actual implementation makes this effort all the more necessary. *Most theories which sound plausible do not work.* AI theories go through radical changes in the implementation process. These changes have nothing to do with the idiosyncrasies of implementation, but result from flaws in the original theory which only become obvious in the implementation process. (See McDermott [76a] for a longer discussion of this.) Throughout this research the program was run on scores of examples to determine the precise points at which the theory had to be extended. In summary, the purpose of writing a program is to debug the theory, not the implementation.

This study of electrical engineering serves primarily as an effective means to an end, and is not the end itself. It is a well-studied discipline. It provides a variety of different structural forcing functions. The circuit schematic provides a formal unambiguous representation for the circuit. Network theory can completely specify the behavior of any circuit. Electrical engineering also has a well-developed taxonomy of abstractions and teleological concepts. A great deal of literature exists on the subject which sheds light on engineers' reasoning processes. Any theory of informal reasoning must relate to these constraints. Indeed most of these constraints originate from informal concepts. Consider feedback, which has a precise and formal definition, yet it is often used informally. Any theory of qualitative reasoning must account for the relation between these two uses of the concept of feedback.

This thesis does not present any rigorous psychological experiments which either motivate or

verify the ideas discussed. The central source of insight has been introspection and observation. As a teaching assistant I had the opportunity to teach circuit theory and observe how students learn electronics. At M.I.T. we have also studied how expert electrical engineers reason about circuits [Sussman & Stallman 75]. Another source of information is the kinds of difficulties students have in understanding circuits. Experience with the SOPHIE project [Brown *et.al.* 74] provided data on the kinds of bugs students exhibit in trying to troubleshoot power-supplies. My research style has been to start with a simple mechanism suggested by these observations, to crystallize it via an implementation, and to push it as far as possible to see how many of the observations it accounts for. Only when absolutely necessary is the simple mechanism extended.

1.5 The Engineering Problem Solving Project at MIT

The Engineering Problem Solving Project (EPSP) is concerned with uncovering the fundamental mechanisms involved in the kind of reasoning employed by people in the design, analysis, debugging and explanation of complex systems constructed to perform a specific function. The first achievement of this project was a formalization of the intuitive notions engineers employ in analyzing circuits quantitatively [Sussman & Stallman 75]. This theory, called propagation of constraints, led to a sequence of increasingly more sophisticated analysis programs, all called EL [Stallman & Sussman 77]. This was accompanied by a fault localization system WATSON [Brown 76] and a circuit design system DESI [McDermott 76b]. Although neither WATSON nor DESI worked as well as EL, this research argued persuasively for the role of teleology and abstraction in understanding circuit behavior.

We are currently working on a longer term project to construct a working design system [Sussman 77a]. We have recently made progress on the less ambitious goal of circuit synthesis [de Kleer & Sussman 78]. Synthesis is the determination of the parameters of the parts of network given desiderata on the network as a whole. Synthesis, unlike the full-scale design problem addressed by McDermott, presumes that the original circuit topology is given. SYN is a working system which can be of assistance to an engineer in the synthesis of a wide class of circuits. SYN's current difficulties result from its inability to understand how the circuit works. In contrast, the failure of WATSON and DESI can be traced to their inability to analyze the behavior at a specific enough level of detail. QUAL analyzes circuit behavior at a deeper level of detail than WATSON or DESI, but at a shallower level of detail than SYN. It addresses the less ambitious

goal of recognition, but strives for robustness.

The mechanisms that QUAL uses for its causal reasoning bear close resemblance to the propagation of constraints technique used in EL and SYN. In fact, the two systems share Lisp code. QUAL records explicit dependency information about its deductions as suggested by TMS [Doyle 77]. The idea of recording dependencies explicitly came from the original EL.

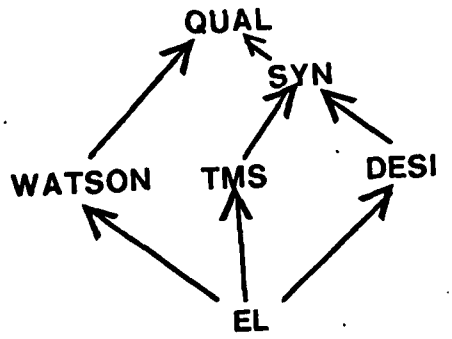


Figure 5 : Evolution of EPSP

Another interest of the EPSP is programming apprentices [Rich & Shrobe 78] [Waters 78] [Shrobe 79]. Many of the ideas underlying the programming apprentice research originated from electronics, and it has progressed to the point where it now has much to contribute to the electronics side of the project. The programming apprentice project has pushed the ideas of teleology and abstraction to far more precision in its attempt to produce a taxonomy of the plan types used in programming. This ongoing work on programming has had considerable impact on this thesis.

The idea that causality and teleology are important in understanding circuit behavior is not new. Brown, McDermott and Sussman have all argued for it. The previous research emphasized the goals of analysis, design and fault localization at the expense of the underlying descriptions of circuit mechanisms. Causality and teleology are very broad concepts representing clusters of distinct ideas. QUAL deals with only a small subset of what Brown and McDermott termed causal or teleological. *What is new is that a particularly simple type of teleology and causality interacting in limited ways is sufficient to account for much of circuit understanding.*

1.6 Other Related Work

Related work on electronics was covered in the previous section. My earlier work on mechanics [de Kleer 77b] is a precursor for this work on electronics and is discussed in chapter 3. One group of related research is characterized by Rieger & Grinberg [77] and Freiling [77] who model human causal reasoning. Another group studies human reasoning of physical systems in more generality without necessarily focusing on causality. People in this group are Bundy *et.al.*[76], Hayes [78] and McDermott & Larkin [78]. QUAL falls between these two groups by demonstrating how causality affects reasoning generally.

My research differs from this related work by focusing on the distinction between the object that manifests the behavior and the abstract mechanism by which the object achieves that behavior. This distinction solves many of the difficulties of Rieger's and Freiling's theories. With the distinction between object and mechanism the theory of causal reasoning and teleological reasoning can be tested by recognition. Without the object, all of the possible causal interactions have to be included by the researcher, thus making the representation highly suspect. Since it is the task of recognition to identify all possible causal interactions, this provides a forcing function on the causal representation. The only way to obtain mechanism from object is by modeling. A model for a component describes its behavior from a particular point of view. Since neither Rieger nor Freiling explore modeling, the content of their mechanism representation is completely arbitrary.

My central objection to Rieger's theory is that it has no structural representation of the device that the cause-effect representation is a description of. This objection raises serious questions about the nature of his theory and is a major source of the theory's difficulties. The structure of a cause-effect diagram for a device is determined by the person who constructed it. Different people will come up with different diagrams and there is absolutely no way to compare them. Further, his representation is nonhierarchical and therefore has little facility for describing mechanisms at a shallower level of detail.

The absence of an object-device results in problems with equality. In Rieger's cause-effect diagram for the forced air furnace there are two states labeled "mercury not at D" and "mercury at D." Since there is no underlying object "D" to refer to, these labels have no meaning. In order to express that these two states are antithetical a separate "state-antagonism" node must

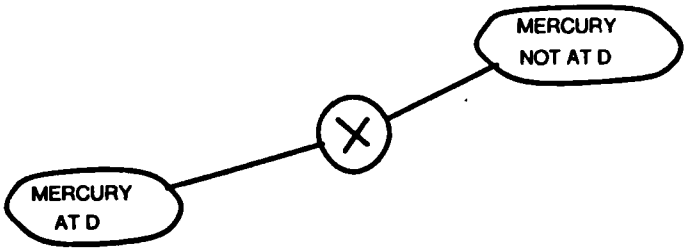


Figure 6 : State-Antagonism in Forced Air Furnace

be placed between them (figure 6). The lack of an object-device makes it difficult to determine whether two actions are interacting since there is no common way of referring to the device parts manifesting the behavior. Rieger utilizes a procedural simulation of the declarative process description to determine unexpected sources of causal interaction. This is an obscure way of temporarily creating an approximation to the object-device.

Rieger does not use his cause-effect diagrams for anything other than simulating the physical devices. He does not explore other ways reason about them. It is a thin horizontal slice of a plausible theory of human reasoning about causality missing any forcing function. Currently, Grinberg [78] is applying Rieger's theory to design, and the application of this forcing function will likely lead to the incorporation of a more explicit notion of the object-device.

Freiling has extended Rieger's work to deal with many of the above objections. His representation is hierarchical thus allowing reasoning to take place at different levels of detail. Rieger's representation does not distinguish between cause and intention, whereas Freiling's does. The combination of hierarchy and explicit progress variables allow Freiling to circumvent the use of an object-device. Freiling explains how his representation might be used for recognition. He defines recognition as moving up the hierarchy from a basic causal description of the device to a more abstract description of the causality of the object-device. Although not the focus of his work, he also discusses the problem of determining the basic causal description of the mechanical device from its geometry. Unlike electronics where a detailed representation of the object-device has been developed by electrical engineers, vision research as not progressed as far thus making the determination of the basic causal description rather difficult.

QUAL utilizes the circuit schematic as an explicit description of the object-device. From this object-device it constructs a representation of the mechanism. This representation is closer to

Freiling's than Rieger's in that it distinguishes cause and intention and in that it is hierarchical. However, QUAL nowhere contains a representation identical to Freiling's or Rieger's. The content of their representations is separated into a number of distinct structures in QUAL. The unique focus of my research is the determination of the function from the structure of the object. This is the problem of recognition.

Chapter 2

OVERVIEW AND BACKGROUND

2.1 The Overall Recognition Process

This chapter presents an overall perspective of the recognition process discussed in chapters 4 through 8. The discussion in this chapter makes some reference to electronics, but does not require the sophistication with electronics that is needed to appreciate those chapters. Chapter 3 discusses the theoretical foundations underlying this recognition theory, and requires no knowledge of electronics.

The task of recognition is to determine, from a description of the circuit, a description of the mechanism by which the circuit achieves its behavior. Recognition is a convenient forcing function since both descriptions for circuits and descriptions for mechanisms are fairly well agreed upon in electrical engineering. A description of the circuit consists of a schematic, and a description of the mechanism explains how each component's individual behavior contributes to the overall behavior of the system. This type of functional description is central for analysis, design and troubleshooting.

In order to recognize an object its properties must be related to those the recognizer is familiar with. One recognition technique, topological analysis, compares the topology of a new circuit with previously recognized topologies. Another recognition technique, functional analysis, determines the behavior of the overall circuit by combining the behaviors of the individual components. Both functional and topological analysis construct a hierarchical description of the circuit. In functional analysis this hierarchy is in terms of fragments of behavior, while in topological analysis this analysis is in terms of fragments of topology. A third technique, geometric analysis, relies on the tacit graphical language engineers use when they describe circuit topologies on paper. Geometric analysis is incomplete by definition, and its only utility is efficiency. Functional analysis is the most powerful recognition technique since the ultimate test of a circuit is whether it functions correctly, and not whether it has the correct topology or geometry. Furthermore, any geometric

or topological analysis must eventually produce a description of the circuit's mechanism.

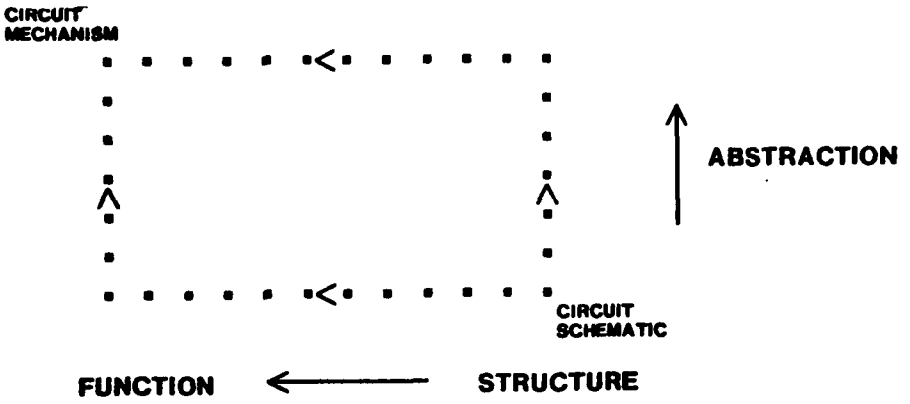


Figure 1 : Functional vs. Topological Recognition

These issues can be illustrated by an example from engines. A mechanic recognizes an automobile engine simply by its shape. However, if asked to explain why it is an engine he will give a functional description of why it operates. When presented with a new engine type (e.g. from a ship or airplane), he will not be able to recognize it with the shape clues for automobile engines but will need to analyze its functioning in order to determine that it is an engine. (In order for him to successfully analyze its functioning he must, of course, recognize the parts of the new engine.)

Functional analysis has theoretical advantages over topological and geometric analysis. The same system component can have multiple purposes and may thus be shared among many modules. (See Steele & Sussman [78] for a discussion of almost-hierarchical systems.) Therefore, any hierarchical description of the system's purpose will be tangled and difficult to reason about. In functional analysis, this undesirable sharing can be isolated to one level of the analysis. If the same component is contributing in two different ways to the system's behavior, the causal and teleological analysis and will discover this and ascribe two primitive behaviors to it, and primitive behaviors are never shared. For example, functional analysis is not confused by the fact that the wheels both support the car and are part of the drive train, because these are two very different behaviors. Topological analysis has to insert "wheel" in two different places in a topological hierarchy.

This can be viewed yet another way. The basic theme of recognition is the determination of mechanism from structure. Topological analysis attempts to produce a hierarchy of structures, each of which has a known behavior. Functional analysis, on the other hand, produces a very primitive description of the mechanism from the structure, and then constructs a hierarchical description of this mechanism, ignoring the original structure.

QUAL's recognition process is summarized by the following flow chart:

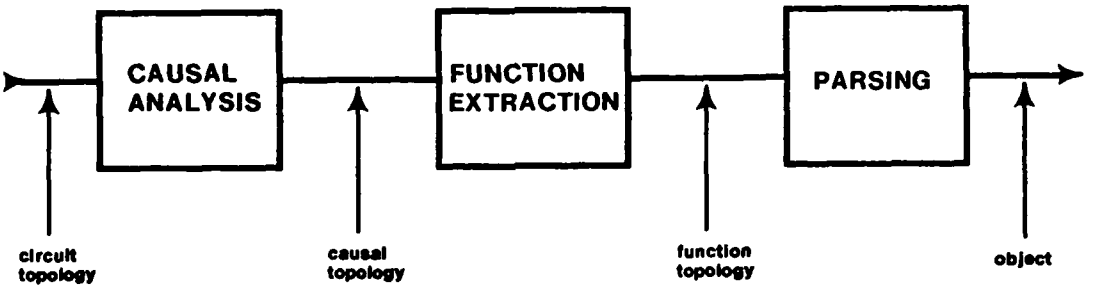


Figure 2 : QUAL's Recognition Process

The following three sections discuss the three major steps in the process.

For this process to be useful and examinable, each step must construct extensive explanations for the circuit's behavior from its perspective, as well as recording reasons for its deductions. Therefore, although the process terminates with a single token describing the system (e.g. amplifier) the user and other programs will have access to why that circuit is what QUAL claims it is.

The process encounters choice points where ambiguities have to be resolved. For the types of circuits QUAL recognizes, these ambiguities can be dealt with by a variety of heuristics. Since these are heuristics, the resulting explanations that QUAL produces are rationalizations.

2.2 Causal Analysis

Causal analysis takes a description of the circuit's topology as an input and produces a qualitative description of the circuit's incremental behavior as an output. The input description includes an annotation identifying the circuit's input-output ports:

```
(circuit: ce-feedback
 nodes: (vcc ground b1 c1 output e2 fp)
 devices: ((q1 (npn-transistor emitter: ground base: b1 collector: c1))
           (q2 (npn-transistor emitter: e2 base: c1 collector: output))
           (rc1 (resistor vcc c1))
           (rc2 (resistor vcc output))
           (rb1 (resistor e2 fp))
           (rb2 (resistor fp ground))
           (rf (resistor fp b1))
           (input (terminal b1))
           (output (terminal output))
           (common (terminal ground))
           (supply (battery vcc ground)))
 input:    (current input)
 output:   (voltage output ground))
```

Figure 3 : Description of CE-FEEDBACK for QUAL

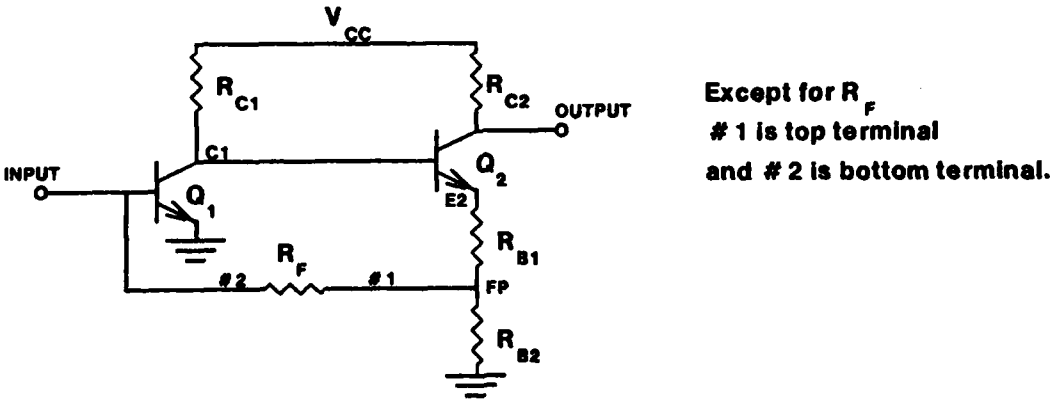


Figure 4 : CE-FEEDBACK

Causal analysis determines the behavior of the circuit by propagating the input through the circuit and constructing a description of the resulting behavior. In the case of CE-FEEDBACK, QUAL produces a description which corresponds to the following english text: "The increased input voltage turns Q1 on harder, pulling down its collector. This falling voltage is applied to the base of Q2, causing it to begin to turn off. Since Q2's collector current is dropping, the voltage

across the load RC2 must also drop." QUAL quantizes each electrical quantity into increasing(⇑), decreasing(⇓) or unchanging(0). Part of the description which QUAL provides is:

```
Starting with input:
(VOLTAGE B1 GROUND) = ⇑
   Premise
```
An increasing input voltage is applied to the circuit.
```
(CURRENT C Q1) = ⇑
   V ⇒ IC for Q1
```
The convention is that currents flow into devices away from nodes. This statement indicates that the current flowing into the collector of Q1 is increasing.
```
(VOLTAGE C1 GROUND) = ⇓
   KCL-heuristic at C1
```
The potential at Q1's collector drops.
```
(VOLTAGE E2 C1) = ⇑
   KVL-heuristic at Q2
```
Since (VOLTAGE <n1> <n2>) represents the voltage from <n1> to <n2>, this is equivalent to (VOLTAGE C1 E2) = ⇓
```
(CURRENT C Q2) = ⇓
   V ⇒ IC for Q2
```
The current flowing into the collector of Q2 is decreasing.
```
(CURRENT #2 RC2) = ⇑
   KCL for node OUTPUT
```
The current flowing into the bottom terminal of RC2 is increasing.
```
(CURRENT #1 RC2) = ⇓
   KCL for device RC2
```
The current flowing into the top terminal of RC2 is decreasing. Two currents appear for RC2 because currents are defined for terminals and not for devices. Ohm's law is specified using the current in the #1 terminal of resistors.
```
(VOLTAGE OUTPUT VCC) = ⇑
   ⇔ V I for RC2
```
Ohm's law for RC2.
```
Also given that:
(VOLTAGE VCC GROUND) = 0
   SUPPLY
```
Since the voltage between VCC and GROUND is fixed by the battery, it cannot change.

```
The combination of events (VOLTAGE GROUND VCC) (VOLTAGE OUTPUT VCC) cause:
(VOLTAGE OUTPUT GROUND) = ⇑
   KVL applied to nodes OUTPUT VCC GROUND
```

There are four important points to bear in mind when reading causal arguments. First, the values refer to changes in circuit quantities and not the quantities themselves. The fact that the collector current is negative or positive bears no relation to whether it is increasing or decreasing. The incremental voltage between the nodes of a battery is always zero since the battery fixes this voltage. Second, the values refer to changes in circuit quantities, not changes in absolute values of circuit quantities. Thus a change of value from -3 to -2 is considered an increase just as a change of value from 2 to 3. Third, the statement "x is increasing" is equivalent to "$-x$ is decreasing." Thus, `(VOLTAGE <n1> <n2>) = - (VOLTAGE <n2> <n1>)`. Fourth, all currents flow into devices away from nodes. These issues are critical to the nature of causal arguments and will be examined in detail in Chapter 4.

All of the standard electrical device models have been reformulated to deal with these qualitative quantities. For example, Ohm's Law $V = IR$, when quantized is $V \Leftrightarrow I$ indicating that the change in V must be the same as the change in I. Since the resistance R is presumed to be positive, its precise value never contributes to the circuit's behavior.

These rules are insufficient to deal with CE-FEEDBACK, and thus QUAL incorporates heuristic rules which it applies when the basic rules break down. From a network theory viewpoint these rules are invalid, but they are extremely useful in analyzing circuit behavior. These rules were discovered by examining the arguments of electrical engineers. Only two heuristics are necessary to deal with most circuits: KCL- and KVL-heuristics. (The heuristics are named after the corresponding Kirchoff's voltage or current law.) The KCL-heuristic predicts that the voltage at a node will drop if the current drawn from the node is increasing (and correspondingly, if the current is decreasing the voltage will rise). For example, since Q1 is pulling current out of node C1, the voltage at C1 drops, even though the currents from RC1 and Q2 are unknown. By the phrases "current drawn from the node is increasing" and "pulling current out of node" I mean that with the sign convention chosen such that current flowing out of a node is positive, that the particular current in question is becoming more positive. This raises a seeming contradiction. For the sake of argument assume that the base current of Q2 is zero. Then the current in RC1 is the same as that of the collector of Q1. Thus the change in current flowing out of the node

into Q1 is equal and opposite the change in current flowing out of the node into RC1. These opposite currents, by the KCL-heuristic, predict opposite voltages at node C1. This contradiction is resolved by introducing the notion of causality: since the current in Q1 "causes" the current in RC1, it is the correct current to use in the KCL-heuristic. The KVL-heuristic predicts that the potential at a device's terminal can be applied directly to the device. For example, the rising potential at the base of Q2 causes its base-emitter voltage to increase. The KVL-heuristic also requires a notion of causality, but this discussion is left until later.

These heuristics make assumptions about the behavior of the rest of the circuit, and as a consequence the causal analysis may discover multiple conflicting behaviors for the same circuit. Both heuristics assume the circuit fragment connected to the node or device is behaving as a positive resistance. For example, the KCL-heuristic assumes that other currents flowing into the node have no effect, and the KVL-heuristic assumes that the voltage at the device's other terminals can be ignored. The causal analysis therefore produces a number of possible behaviors, or *interpretations*, for the circuit's behavior.

QUAL summarizes the causal argument by a *mechanism graph* which describes how the circuit achieves its behavior:

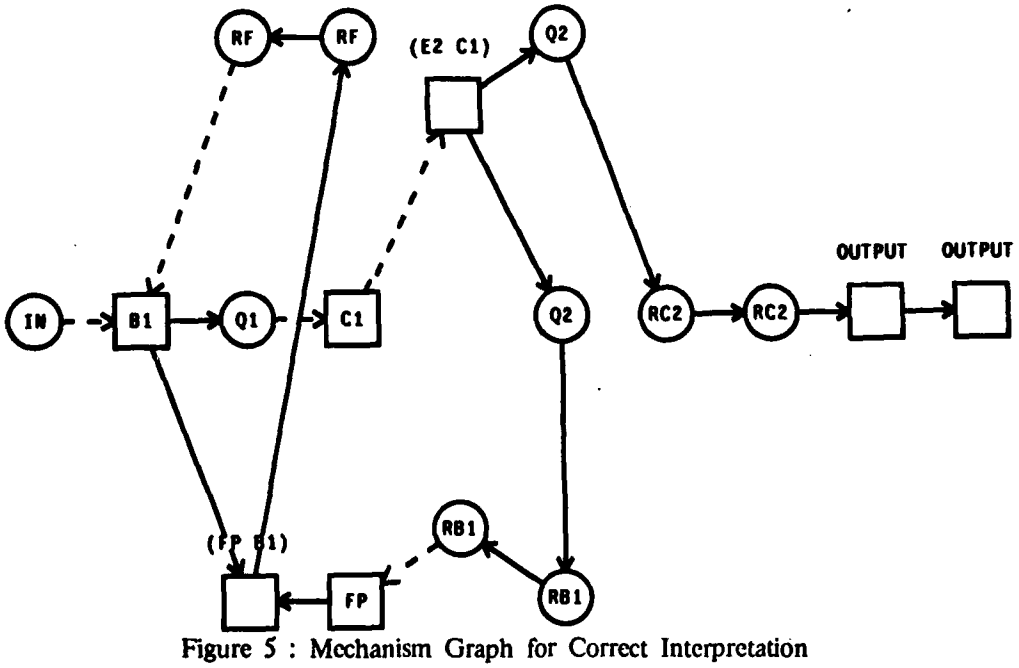


Figure 5 : Mechanism Graph for Correct Interpretation

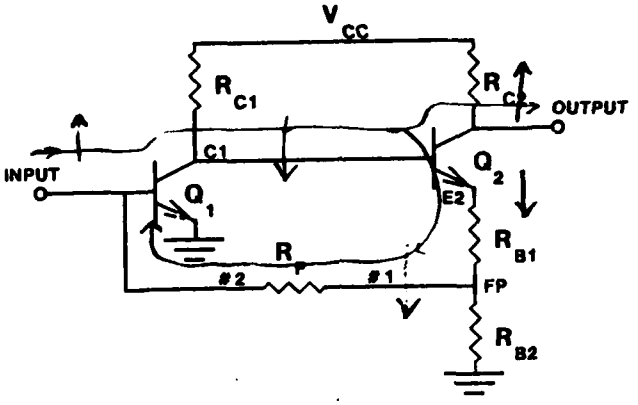


Figure 6 : CE-FEEDBACK

Every vertex of the mechanism graph corresponds to a changing circuit quantity. Voltage vertices are indicated by squares and are labeled by the voltage's two nodes. If one of the nodes is incremental ground, the vertex is labeled with the name of the remaining nonground node. Current vertices are indicated by circles. Since currents only flow through terminals, these vertices are labeled by the terminal's device. Every edge of the mechanism graph corresponds to the application of a causal rule. If the causal rule makes an assumption, the edge is indicated by a dashed instead of solid line.

The path through the mechanism graph IN → B1 → Q1 → C1 → (E2 C1) → Q2 → RC2 → RC2 → OUTPUT → OUTPUT is in one-to-one correspondence with the causal argument presented at the beginning of the section (after figure 4). The remaining edges and vertices describe the feedback path.

2.3 Function Extraction

For each of the interpretations produced by the causal analysis, the function extraction phase constructs a description in terms of the behavioral features used by electrical engineers. Finally, it chooses that interpretation which exhibits the most plausible features.

The electrical models utilized within the causal analysis characterize the device's behavior in every possible context. However only a part of this general description is required to deal with the behavior of the device in any particular interpretation. This part characterizes the *function* of the device with respect to the interpretation. Since function describes how individual devices contribute to the overall behavior of the system, it assigns purpose to the individual devices. In QUAL this simple kind of purpose description is called *implementation teleology.*

Consider CE-FEEBACK again. The causal argument for CE-FEEDBACK indicates that Q1's input signal is applied to its base and that its output signal is its collector current. This is known to engineers as the common-emitter configuration. Similarly, Q2 is also functioning in the common-emitter configuration. (There are 11 such transistor configurations. The resistor has 6 configurations.) For example, RC2 is functioning as a current-to-voltage converter. QUAL produces the following description of the implementation teleology of CE-FEEDBACK:

```
Q1 is functioning in the common-emitter configuration.
Q2 is functioning in the common-emitter configuration.
Q2 is functioning in the common-collector configuration.
RC1 is functioning as a current load.
RC2 is functioning as a current load.
RB1 is functioning as a wire.
RB2 is functioning as a current load.
RF is functioning as a voltage sensor.
```

Note that Q2 is functioning in two configurations: in the common-collector configuration on the feedback path and in the common-emitter configuration on the main signal path. Because causal analysis does not adequately deal with bias, it cannot find any meaningful purpose for RB1.

Each configuration shares inputs and outputs with other configurations permitting the configuration topology to be represented by a graph similar to the mechanism graph:

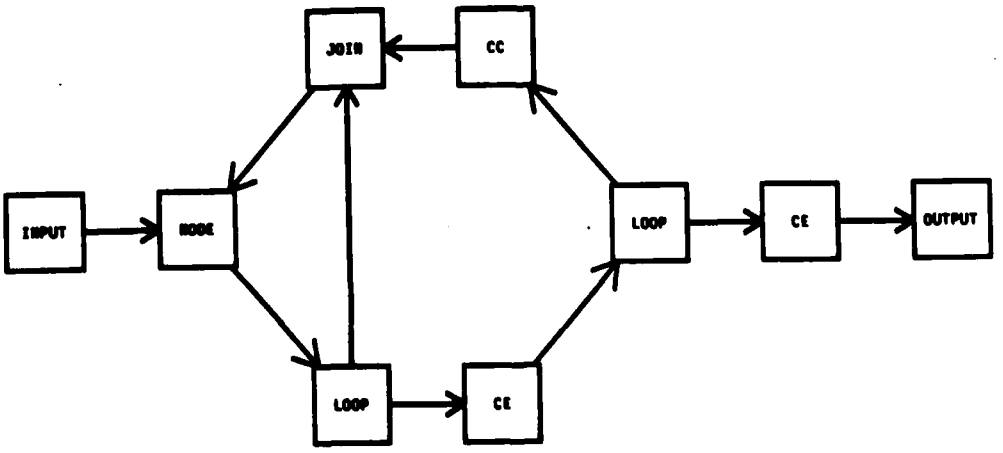


Figure 7 : (Abbreviated) Configurations for CE-FEEDBACK

The vertices correspond to configurations and edges indicate shared input-output ports. In figure 7, every edge corresponds to some vertex of the mechanism graph. The vertex labeled NODE indicates that B1 is a comparison point of type node. The vertex labeled LOOP indicates that Q2 is a sampling point of type loop.

Electrical circuits have been studied extensively and as a consequence, the potentially useful configurations have been culturally identified (although somewhat informally). QUAL's library contains a taxonomy of these configurations, all of which can be identified by connected patterns of vertices in the mechanism graph. The construction of the configuration graph from the mechanism graph is therefore straightforward.

Each interpretation leads to a different implementation teleology, and QUAL chooses that interpretation which assigns maximum purpose to the components, as the correct one. For example, in the interpretation in which the signal flows through RF to Q2 bypassing Q1, Q1 has no purpose and therefore the interpretation is rejected. The correct interpretation can be determined *without* knowing the ultimate purpose of the system. One of the reasons this strategy is successful is that circuits are designed to meet minimum cost constraints and therefore only components with functional purpose are ever included.

2.4 Parsing

The final step of the recognition process takes the configurations produced by the function extraction step and produces a hierarchical description of how the functioning of the individual components contributes to the overall behavior of the circuit. This step determines that CE-FEEDBACK is a two-stage feedback amplifier with Q1 being the first stage and Q2 the second, and that both stages exhibit voltage and current gains such that the overall amplifier has high voltage and current gain.

Each piece of behavior is represented by a *fragment*. The configurations provided by the function extraction step form the primitive fragments which are parsed with topological rewrite rules. In the parsing step these rewrite rules are applied until only one fragment remains, thus producing a hierarchical description of the circuit's behavior. The simplest rewrite rule is the cascade rule:

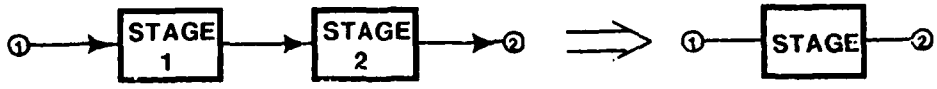


Figure 8 : Cascade Rewrite Rule

The gain of the composite is the product of the constituent gains. The input impedance of the composite fragment is the input impedance of the input stage and the output impedance of the composite fragment is the output impedance of the output stage. QUAL's library contains a grammar of approximately 30 such rules which are applicable to power-supplies and amplifiers.

Since these rules deal with function and not structure there is no sharing, and the description produced by the rewrite rules is completely hierarchical. QUAL explains the purpose of each component by listing its parent fragments within the hierarchical parse:

```
Q1 is functioning in CE configuration.
Which is STAGE1 of CASCADE
Which is BASIC-AMPLIFIER of FEEDBACK
Which is STAGE of TOP-LEVEL

Q2 is functioning in LOOP configuration.
Which is SAMPLING of FEEDBACK
Which is STAGE of TOP-LEVEL
And,
Q2 is functioning in CE configuration.
Which is STAGE2 of CASCADE
Which is BASIC-AMPLIFIER of FEEDBACK
Which is STAGE of TOP-LEVEL
And,
Q2 is functioning in CC configuration.
Which is FEEDBACK-COUPLING of FEEDBACK-NETWORK
Which is FEEDBACK-NETWORK of FEEDBACK
Which is STAGE of TOP-LEVEL
```

```
RC1 is functioning in I-LOAD configuration.
For Q1 functioning in CE configuration.
Which is STAGE1 of CASCADE
Which is BASIC-AMPLIFIER of FEEDBACK
Which is STAGE of TOP-LEVEL
```

The topological rewrite rules have much the same structure as the rewrite rules that might be used in the topological analysis of a circuit schematic. However, these rules operate on the topology of the causal interactions, not the topology of the physical connections. Besides the theoretical advantages discussed in section 1, this strategy abstracts away much of the surface details of the circuit topology and provides a simple kind of canonicalization which makes the last step of the recognition process much easier.

2.5 *Propagation of Constraints Applied to Circuit Analysis

When the engineer needs to know the detailed behavior of the system, he models the behavior of the components as constraint equations and manipulates these equations in order to solve for the quantities of interest. Classical network theory has a formal technique for constructing a necessary and sufficient number of these constraints: node equations or loop equations. The resulting simultaneous equations are guaranteed to be solvable for the quantities of interest. However, we have observed that few engineers use these techniques to solve circuits. Instead they employ a tacit calculus for constructing and manipulating these constraints which takes advantage of the idiosyncratic structure of equations about electrical quantities and minimizes the amount of symbolic manipulation that is required to solve them. This tacit calculus has been articulated in recent AI research, and is called *propagation of constraints*. This section discusses propagation of constraints and two circuit analysis programs based on it, since they are the precursors to the research presented in this report.

Propagation of constraints is directly related to the theory of causal reasoning developed in chapter 4. Although EL and SYN are capable of analyzing and synthesizing a variety of circuits, they suffer from a number of difficulties whose origin can be traced to the incomplete characterization of the calculus engineers use to analyze circuits. The theory of causal and teleological reasoning presented in this thesis is one part of this tacit calculus that is missing from EL and SYN. As this theory is developed, I will explain how it solves some of the problems of

EL and SYN.

SYN, the latest of a sequence of circuit analysis programs based on propagation of constraints, was developed specifically for circuit synthesis. Synthesis, the determination of the parameters of the parts of a network given desiderata for the behavior of the network as a whole, is a major facet of design. However, the following discussion only considers its analysis capabilities.

Abstractly, a circuit is made of cells, each of which represents an electrically interesting quantity, such as a voltage, current or resistance. A cell may participate in one or more constraint expressions each of which represents an electrical circuit law. A constraint expression involves several cells, thus the voltage across a resistor, the current through it, and its resistance are related by a constraint expression which is an instance of Ohm's law for that particular resistor.

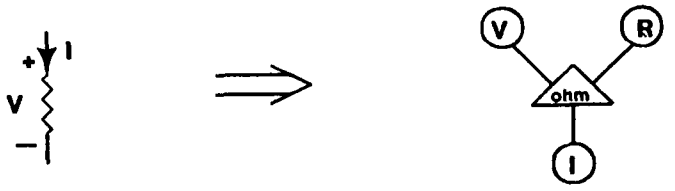


Figure 9 : A Resistor as a Constraint Diagram

When a model of a circuit is made, a network of cells and constraint expressions is constructed. For example the following circuit

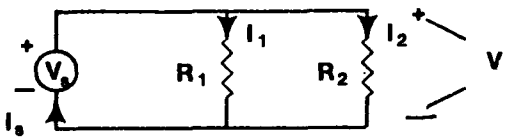


Figure 10 : A Simple Circuit

may be represented by the following simplified constraint diagram. (SYN's constraint diagram is more complex.):

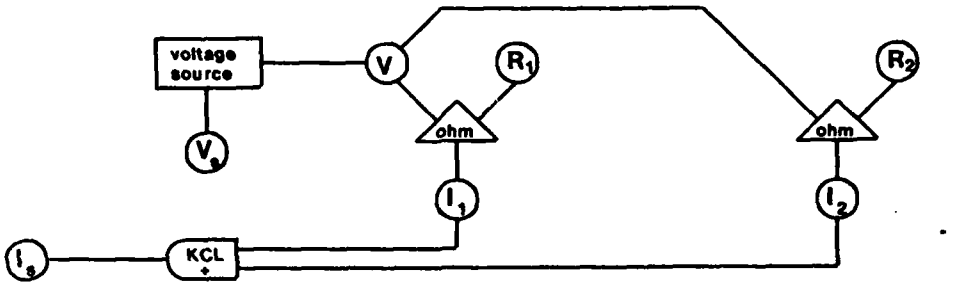


Figure 11 : A Constraint Diagram for the Circuit of Figure 10

Each cell may have a value. The value may come from the user or it may be deduced from other values by constraint expressions. When a cell is assigned a value each constraint it participates in is considered to determine if enough information is available to enable it to use that constraint to deduce a value for another cell. Discovering a new value may thus determine yet other values, thus "propagating the constraints."

Sometimes two different constraints each can produce a value for the same cell. If this condition, called a *coincidence*, occurs, the values must be the same for the set of constraints to be satisfiable. If the values are constants, and if they are equal no new information is deduced, but if the constants differ, a contradiction has been found. A contradiction indicates that some faulty assumption has been made in the analysis. Sometimes, the value is a symbolic expression. In the case of a coincidence equating symbolic quantities, there is a third possible outcome. One symbolic quantity may be eliminated by solving for it in terms of the others.

Consider an example: SYN knows the voltage of the source V_S, the resistance R_1, and the resistance R_2 in the circuit of figure 10. Looking at the constraint diagram in figure 11, we see that the only constraint which can make a deduction is the voltage-source law. Thus cell V is assigned a value equal to the value of cell V_S. The constraints attached to cell V are now examined to determine if any other deductions can be made. Ohm's law for both resistors can combine the values of V and their resistances to produce values for their currents. I_1 and I_2 can now be combined to determine a value for I_S.

The method does not always work so easily. If in the circuit of figure 10, SYN was told I_S instead of the voltage, V_S, no local constraint expression would have enough information to make any deductions by itself, though the behavior of the network can be totally determined from the given information. The problem involves an inherent simultaneity in the constraints. This can be overcome by introducing a symbolic quantity and propagating it as if it was known. This symbolic quantity is called an *anonymous object* since it is propagated *as if it were known* in the hope that ensuing propagations will constraint its value. In this example, SYN could give cell I_1 the value a. Now it is possible to use KCL to deduce that I_2 is $I_S - a$ and it is also possible to use Ohm's law to deduce that V is aR_1. These new values can be further propagated. Using Ohm's law on the other resistor allows us to determine that V is $(I_S - a)R_2$. But SYN already knows a value for V. Hence there is a coincidence, and the algebraic equation:

$$(I_S - a)R_2 = aR_1$$

must be solved. It can be solved:

$$a = \frac{R_2}{R_1 + R_2} I_S$$

The value of V is now known in terms of given parameters. This value can be propagated by the voltage-source law to give a value for V_S.

Such examples illustrate the need for symbolic algebraic manipulation in a program which performs analysis by propagation of constraints. The simultaneity is apparent in the constraint diagram (see figure 11) because there is a loop of constraints containing only unknown quantities. In the first example, the loop was broken when V was determined by propagation from V_S outside of the loop. In the second example, a symbolic unknown, a, was used to break the loop. The unknown could have been introduced anywhere in the loop and the coincidence could have happened anywhere in the loop. In essence, propagation is a means of constructing a small, dense set of equations from a large but sparse set.

A typical circuit SYN can deal with is the cascode:

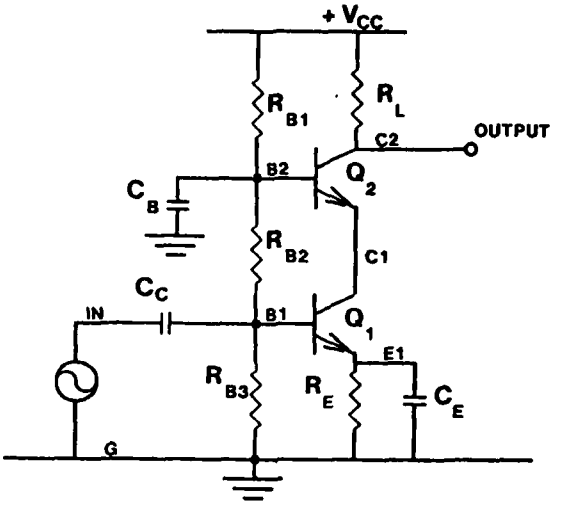


Figure 12 : A Cascode Amplifier

By examining quiescent and incremental models of this circuit, SYN determines that its midband gain is:

$$\frac{r_{\pi 1} g m_2 R_L}{r_{X1} + r_{\pi 1}}$$

In order to solve a circuit by propagation of constraints, all possible constraints relating interesting circuit quantities are constructed. This collection includes all the device rules, applications of KCL at every node and device, and applications of KVL around every loop. These constraints are not represented as algebraic equations, but simply as black boxes which have no internal structure and whose inputs and outputs represent circuit quantities. Each such box represents a relation between circuit quantities that may be useful for solving the circuit, and has the potential of becoming an equation. This collection, or constraint graph, is a representation of all the potential equations that might be created to describe the circuit. Node equations and loop equations can be viewed as a way of choosing a specific subset of the black boxes that produces a necessary and sufficient set of equations to solve the system.

Propagation of constraints is another technique for solving the constraint graph which often requires less algebraic manipulation. It can be best illustrated by the following protocol. Suppose that we are told that the dc bias at B1 is 5 volts:

" ... the voltage at B1 is 5 volts. Since Q1 is on, its base-emitter drop is .6, and therefore the voltage at E1 is 4.4 volts. The currents flowing into the bases are negligible, therefore the same current must be flowing in RL as RE. RE is 4.4k, therefore the current flowing in RE and RL must be 1ma. Since RL is 8.8k and VCC is 15 volts, the voltage at C2 must be 6.2 volts. Since the base currents are negligible, the same currents must be flowing in RB1, RB2 and RB3. Therefore we can determine the voltage at B2 and C1."

This solution by propagation of constraints requires no variables, no algebra, and little arithmetic. A solution by node equations requires 4 equations in 4 variables — no engineer would analyze a circuit in this way and keep his job. The basic idea of propagation of constraints is to fill in the constraint graph starting from what is known expanding into what is unknown. In the above example, this simple propagation results in the discovery of every voltage in the circuit. However, propagation can sometimes get stuck before discovering every interesting circuit quantity and in this case a variable is introduced and propagated as if it were a number. (Another variable is introduced if the propagation gets stuck again.) These variables are the anonymous objects. Therefore, if the voltage at B1 isn't known, a variable can be introduced for it and propagated as if it were known. Consideration of the bias network is sufficient to determine it. For example, suppose RB1=70K, RB2=30K, RB3=50K. Let the voltage at B1 be x. A formal description of the ensuing propagation is as follows:

1. V(B1 GROUND) = x	*Given*
2. I(RB3) = $\frac{x}{50K}$	*Ohm's Law for RB3*
3. I(RB2) = $\frac{x}{50K}$	*KCL at node B1*
4. V(B2 B1) = $\frac{3x}{5}$	*Ohm's Law for RB2*
5. I(RB1) = $\frac{x}{50K}$	*KCL for node B2*
6. V(VCC B2) = $\frac{7x}{5}$	*Ohm's Law for RB1*
7. V(VCC GROUND) = $3x$	*KVL for GROUND B1 B2 VCC*
8. x = 5 volts.	*Solving 3 x = 15*
9. V(E1 GROUND) = 4.4 volts	...

This propagation of constraints solution requires only one variable, while node equations would require 5.

SYN spends most of its resources, both time and space, in algebraic manipulations. The

intermediate expressions it generates in developing a solution are far more complicated than those an engineer would generate. On the cascode problem, some of SYN's intermediate expressions fill up half of a page. (The classical techniques will do even worse.) Although the problems are made manifest in algebraic manipulation, they are not entirely the algebraic manipulator's fault. SYN's problems can be traced to three areas:

1. Failure of the symbolic manipulation routines to take full advantage of the idiosyncratic structure of the expressions.
2. Lack of a theory of where to introduce anonymous objects.
3. Poor choice of device models.

SYN uses the algebraic manipulation routines of MACSYMA [77] to represent and simplify the expressions in the cells. Unfortunately, these routines are incapable of taking advantage of the stereotypical structure of expressions representing the behavior of electrical systems, and this complicates the manipulations more than necessary. This is a subject for further research. QUAL addresses the second two difficulties. A poor choice of anonymous objects can lead to excessively complicated expressions. Even worse, a poor choice of anonymous object makes the explanation SYN generates for its deductions incomprehensible. The engineer prefers to see the intermediate expressions in terms of meaningful quantities. For example, in the cascode problem he prefers to see intermediate expressions in terms of V_{BE1} and not the current in the battery (the quantity SYN usually chooses). The solution to this problem comes out of the technical details of causal analysis, and is presented in section 6.9. Since SYN does not know how any device is being used, it must use the most precise model for each device. QUAL, with its knowledge of the role each device plays in meeting the ultimate purpose of the circuit, can advise SYN on what model to chose. This is discussed in section 7.10.

Chapter 3

THEORETICAL FOUNDATIONS

3.1 Chapter Outline

This chapter includes an overview of the basic ideas of the thesis from a nonelectrical perspective. I draw my examples from elementary physics and mechanical devices. As each idea is presented, its application to electronics will be alluded to, but not discussed in detail. The claims and arguments made in this chapter are not meant to be persuasive — that is left for the following chapters which apply the ideas to electronics.

3.2 Points of View and Levels of Detail

A goal of this research is to identify the calculus that humans tacitly employ to reason about physical systems. Although I claim that the ideas presented here have widespread applicability, they have been worked out to their greatest extent for designed artifacts, and in particular, electronic circuits. To identify the calculus, I use the tools of artificial intelligence to construct a model for this tacit reasoning. The building of an expert problem-solving system is not an immediate goal of this research, although I believe that this approach will aid the development of expert problem-solving systems.

In solving a problem, a competent human will utilize the simplest strategy possible. As well as being able to solve difficult problems, he solves simpler problems with qualitatively simpler techniques. The human problem-solver appears to utilize different representations for the same problem, and communication between these representations provides a framework for representing knowledge and for guiding the problem-solving process. The two most common simplification techniques are abstraction and the imposition of a particular point of view.

Abstraction removes irrelevant details by summarizing them with more general concepts. The resulting abstract description is easier to reason about than the original. For example, an automobile engine can be described as working by explosions which move pistons. However,

in order to understand that the automobile is self-propelled it is sufficient to know that the engine produces the force that moves the car. The concept "engine" summarizes all the details such as explosions and moving pistons which make up a functioning engine. The advantage of abstraction is that it lets you describe the system under consideration at the shallowest level of detail necessary to deal with the problem.

Like a car, an electronic system can have thousands of components. Since it seems impossible to consider more than a small number of these components at one time, the engineer collects the components into a few groups whose behavior he understands.

In contrast to abstraction which removes details by summarizing them into more general concepts, taking a point of view arbitrarily removes details which are not relevant to the problem at hand. For example, the garage mechanic may take the point of view that an automobile has an engine with pistons. However, the parking garage designer ignores these details and considers the automobile only as a mass and volume. The scrap iron dealer is interested in the chemical composition of the car so that he can ensure that the recycled iron is of sufficient quality. The advantage of taking a point of view is that it lets you consider the system as a simpler object whose behavior is identical to the original for the problem at hand.

Points of view are important in electronics in order to characterize different aspects of a circuit's behavior. For example, a circuit can be analyzed from an ac or dc point of view. Circuit behavior can usually be understood by considering the different points of view independently. This is not true for designer, who must consider all the points simultaneously in order to trade off the desiderata in one point of view with that of another.

The ideas of abstraction and point of view are not new. They are central to engineering, the science of designed artifacts. They are mentioned here because they are parts of the foundation of the calculus humans tacitly use.

3.3 Envisioning and Rationalization

The fundamental tacit calculus humans use to understand physical systems is *envisioning.* Envisioning is a qualitative simulation of the system under study which constructs a primitive description of the behavior of the system. This primitive description plays a central role in any deeper understanding of the system. The simulation is accomplished by quantizing the state of the system into its important regions and modeling the behavior of the components by rules relating

the state of each component with the states of its neighbors. The description of the behavior of the system, or *envisionment*, consists of a sequence of system states with actions between states justified by the component models.

As an example of envisioning, consider the mechanics mini-world of roller-coasters — the kinematics of objects moving on surfaces.

Figure 1 : Will the cart reach *x*?

A description of the envisionment, in English, might be:

"The cart will start to roll down the curved surface without falling off or changing direction. After reaching the bottom it starts going up. It still will not fall off, but it might start sliding back. If the cart ever reaches the straight section it still will not fall off there, but it might reverse its direction."

The program NEWTON [de Kleer 77b] embodies a formal theory of envisioning for the roller-coaster world and is capable of generating a description of the behavior similar to the above protocol. NEWTON characterizes the state of the roller-coaster system by the location and velocity of the cart. The velocity is quantized into upward, downward and zero. The surface itself is divided into segments at the points where there are zeros or discontinuities in the slope. The location is given by the segment where the cart is located and an indication whether the cart is above or below this segment. Thus the surface of figure 1 is described by 3 segments and 4 boundary points. These segments and points are the components of the system.

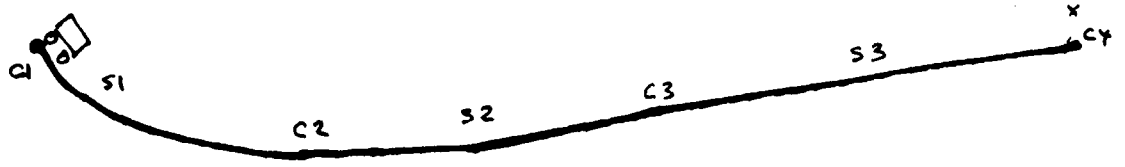


Figure 2 : NEWTON's segmentation

The same collection of rules is used to describe the behavior of the cart on every segment. An analogous collection of rules is needed for points. These two models describe how the components

of the system contribute to its overall behavior. Two rules used in the protocol are: (1) If the cart is on an inclined surface, it may start moving in the downward direction, and (2) A cart may continue moving in its original direction. Eleven such rules are necessary to characterize the behavior of the roller-coaster system. The envisionment of the scene of figure 1 is summarized by the following tree of cart locations.

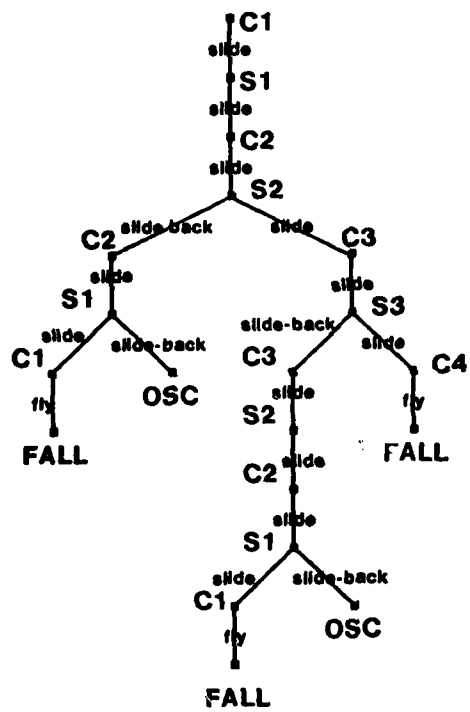


Figure 3 : Envisionment

This representation of the roller-coaster system's behavior has a number of advantages. Simple questions can be answered by simple techniques. For example, it indicates that the cart could reach x and that it will not fly off of the surface. Further, the envisionment also applies to incompletely described scenes, as in the case where the cart will reach C2 no matter what the exact shape of S1 is. Most importantly, the branches in the envisionment tree indicate which ambiguities have to be resolved. The pattern of actions around each ambiguity determines what

mathematical technique should be utilized to resolve it. NEWTON groups the patterns into 5 categories and is capable of mathematically analyzing each one.

The power of the qualitative calculus comes from the fact that it is *complete, limiting* and *articulate.* Envisioning is complete for roller-coasters since it is capable of simulating every possible roller-coaster behavior. Therefore, any behavior which the envisionment does not predict as a possibility cannot happen. Envisioning is limiting in that it generates very few ambiguities. Finally, envisioning is articulate in that it identifies the source of the ambiguities so that other knowledge can be used to deal with them. Without any one of the these properties the qualitative calculus would be useless. For example, without the completeness property no necessary relation exists between the envisionment and what is physically the case. If the envisionment does not identify the source of the ambiguities, other knowledge cannot be used to resolve them. One of the results of this research, which is presented in chapter 4, is an envisioner for electrical circuits that is complete, limiting and articulate.

Since envisioning is limiting, it gives plausible explanations for what actually happens. Suppose the student built a model for the problem and empirically determined that the cart reached x. This is consistent with one path through the envisionment tree. This path is a *rationalization* for the observed behavior since it explains how the behavior could occur, but does not guarantee it. In order to substantiate a rationalization, NEWTON rigorously resolves each ambiguity with mathematical techniques. It is important to note that although envisioning can make algebraic analysis easier, algebraic analysis alone cannot solve many physics problems. Classical mechanics never formalized how to set up the equations of motion. Envisioning is the calculus to do this. For more details see [de Kleer 77b] and [de Kleer 75].

3.4 Causality and Constraint

Electrical engineers appear to use a similar kind of envisioning to reason about circuits. The extension from envisioning for mechanics to envisioning for electronics is not straight forward, but the basic themes remain the same. An obvious problem that has to be tackled in building an envisioner for a new domain is that of identifying components and describing their behaviors. However, electrical circuits posed some unforeseen problems, the solution to which yields a deeper theory of envisioning.

Unlike the roller coaster world, a change in one part of an electrical circuit can have immediate impact on every other part of the system. In order to envision these changes, they have to be ordered. In the roller coaster world each event in the envisionment has a unique antecedent which can be determined trivially. This is not the case in electronics, and the central problem in building an envisioning theory for electronics is the determination of unique antecedents, or *causes*, for events. Thus, an envisioning theory for electronics is a theory of causal reasoning for electronics. A simple mechanical example illustrates some of these issues.

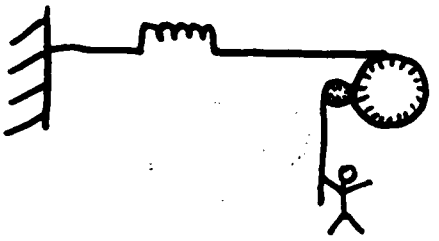


Figure 4 : Spring Mechanism

The man might describe his action of winding up the spring as:

"Pulling on the chain causes the small gear to turn, which causes the large gear to turn, which stretches the spring. As the spring extends it becomes harder to stretch further, which makes the larger gear harder to turn, which makes the small gear harder to turn, which makes it harder to pull on the chain."

Although pulling on the chain immediately results in the whole system changing, we have no trouble assigning a causal order to the changes in the spring mechanism. The extension of the spring and the length of chain the man has pulled down are related by a simple equation:

$$\text{Spring-extension} = \text{Chain-pulled}\frac{\text{small-radius}}{\text{large-radius}}$$

This equation expresses the physical constraints whereby the length of chain pulled is determined by the extension of the spring, and the extension of the spring is determined by the length of chain pulled. In other words, the extension of the spring and the length of chain pulled simultaneously constrain each other. (As do the associated forces and displacements.) The man's causal story bears little resemblance to this constraint, although both describe the same mechanism.

It is fairly easy to construct an envisioner for the spring mechanism since the number of causal interactions are few, and the components of the system are easily identified. For electronics, the difficult problem of identifying the components has been solved, and this solution forms the basis of modern electrical circuit theory — the lumped parameter formulation. However, this formulation is based on constraints, and even for a simple circuit with few components the number of such constraints is very large. Nevertheless, the engineer utilizes envisioning on the lumped circuit model to understand the system's behavior. He can not utilize the causality underlying the constraint system, as was possible in the spring mechanism, since that causality is far too complicated to be tractable. Instead he imposes a mythical causality on the behavior which may bear little resemblance to what is actually the case, but which is simple enough to make envisioning possible.

The following spring network analog of a circuit illustrates some of these issues:

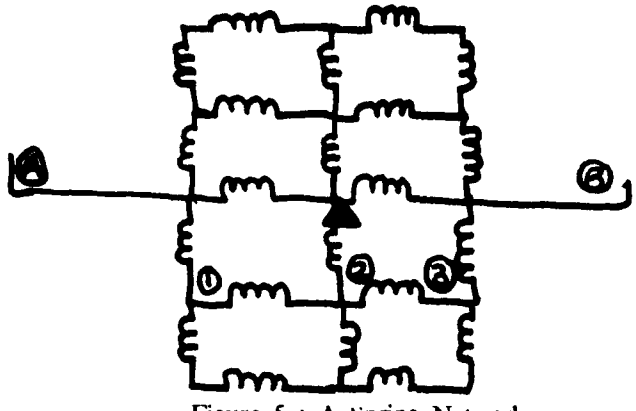


Figure 5 : A Spring Network

In this spring network, the behavior of each node is described by a constraint relating its position to that of its immediate neighbors. When node A is moved, most nodes in the network change position to satisfy the constraints. (In the lumped parameter formulation of an electrical circuit, unlike this mechanical network, these changes happen instantly.) A possible explanation for why B changes position is that node A causes node 1 to move, which causes node 2 to move, which causes node 3 to move, which results in node B moving. This explanation is mythical in that such a path can be constructed from every node in the system, and there is little reason to select the path from 1 to 2 to 3. While such explanations may not be useful for spring networks, the analogous situation arises in electrical circuits where engineers employ such mythical explanations

all the time. Furthermore, every engineer will give a nearly identical causal explanation. Many of the more sophisticated notions of electrical engineering are built on these causal explanations. This unanimity is surprising since the field never formalized this causal reasoning. This is the tacit calculus that this research articulates.

When the envisioning progresses to a point where none of the rules are applicable, it employs two heuristics which enable envisioning to continue. Since the heuristics may predict multiple, differing actions, the envisionment splits at these points. Unlike NEWTON's rules, the electrical device rules always predict a unique resulting action, and only an application of a heuristic results in a possible branch in the envisionment. The result of the envisioning is a number of possible paths through the envisionment tree. Each such path is characterized by a collection of assumptions. These collections are called *interpretations*. As in NEWTON, one of these interpretations is guaranteed to be correct. In fact, I conjecture that every path through the envisionment is achievable with some modifications to the circuit below the level of detail captured by the qualitative models. (The analogous claim does not hold true for NEWTON.)

Envisioning is made possible by describing the behavior of the electrical circuit at a shallow level of detail. Instead of the quantitative constraint rules, the causal qualitative device rules refer only to whether the quantity is increasing, decreasing or unchanging. This is a direct analog to NEWTON which described cart velocities as either upward, downward or zero.

Envisioning is successful since the computation underlying it is finite and robust. The finiteness comes from the fact that the number of possible interactions is known *a priori*, and that the nature of the interactions are strictly controlled by limiting the values that interacting variables can have. In both NEWTON and QUAL the topology of the model of the possible interactions is determined directly from the physical structure of the device. Combined with the fact that the possible values of an interesting quantity are limited to two or three in number, this ensures that the amount of computation involved in envisioning is very small. The limited computation is not achieved by encoding more information into the descriptions of the constituent objects. The objects of the system are grouped into a few classes (NEWTON and QUAL both have about a dozen) and each such component is modeled in exactly the same way, independent of its context. This ensures that the envisioning is robust since it can handle every system which is constructed from these primitive constituents.

The envisionment explanation of a behavior is sequential, with the focus of the sequence

shifting over the physical components of the system. This is independent of whether the system being described functions sequentially (e.g. roller coasters) or in parallel (e.g. electrical circuits). It could be argued that the sequential nature of explanations is an artifact of the linear nature of human communication. However, I believe it is more likely a consequence of the sequential nature of human reasoning (for problems of this complexity). This research does not address itself to this question, and I do not claim that the process by which QUAL constructs the envisionment is identical to the one humans use. I only claim that the result of the process is an explanation very similar to the ones engineers construct, and that these arguments form the basis for more sophisticated reasoning about the system. Indeed there are few mechanisms which can construct the envisionment reasonably simply. For example, if there are no ambiguities in the envisionment, the mechanism must produce the events in sequence (as QUAL and NEWTON do). However, when there are branches in the envisionment, the engineer will move among branches in ways I have not yet been able to precisely characterize. NEWTON and QUAL generate the envisionment depth-first.

One of the reasons the mythology successfully explains circuit behaviors is that the circuit was designed by an engineer using the same calculus. Understanding a circuit corresponds to rediscovering the designer's original intentions, the subject of the next section.

3.5 Teleology

Circuits are systems designed and manufactured to achieve specific functions. Since they have to be conveniently designed, efficiently manufactured and easily maintained, the designer attempts to make his circuits as simple as possible. These desiderata dictate that every component must contribute in some way to the ultimate purpose of the device. This is the *teleological* perspective.

For natural systems such as the kinematics of objects moving on surfaces, the ambiguities must be resolved by quantitatively solving the constraints. For artifacts having specific purposes, such as electrical circuits, knowledge of these purposes can be used to deal with ambiguities. Only the designer needs to rigorously resolve the ambiguities in order to determine precise numerical values for circuit parameters. The correct interpretation is the one that exhibits the intended behavior. The envisioner thus makes a rough test of whether a circuit can possibly achieve its intended purpose.

A selection of the correct behavior by means of quantitatively solving the ambiguities constitutes a proof of the behavior. The interpretation selected on the basis of teleological evidence remains a rationalization. However, when an engineer gives an explanation for a circuit's behavior he will accompany the envisionment with a brief commentary describing how a few components contribute to the device's ultimate purpose. This commentary does not give the teleology for every component, but it resolves enough of the ambiguities so that the listener can identify the correct interpretation. Any such explanation can be converted into a proof by rigorously verifying the teleological comments.

Within a particular interpretation, the role of each component can be determined by examining the envisionment. I call this kind of teleology, which makes no specific reference to the system's ultimate teleology, *implementation teleology*. Surprisingly, just the notion that the system has a specific purpose, without knowing what this purpose is, is almost always sufficient to resolve all ambiguities. The correct interpretation is the one which assigns an implementation purpose to the maximum number of components. For example, if you have two explanations for how an automobile moves, one including the engine while the other does not, the odds are that the first explanation is the correct one. Manufacturers usually include only the components that contribute to the designed artifact's ultimate purpose.

There is a second way that implementation teleology helps select the correct interpretation. Electrical circuits have been studied exhaustively, and almost all of the potentially useful combinations of events (called *configurations*) in the envisionment have been categorized. Any interpretation which exhibits a configuration that is not known is probably incorrect. Suppose that in one interpretation of an automobile engine's behavior, the water in the radiator was considered to be the lubricating agent. Although water is a liquid, it is a relatively poor lubricating agent and no matter what the purpose of car, the lubricating agent is probably not water. This technique is successful since the design of engines and circuits is well understood, and a collection of good design rules have been agreed upon. These two criteria usually independently select the correct interpretation. Both of these heuristics depend on the fact that circuits have been studied, and that the sole purpose of the artifact is functional. These heuristics would not apply as well to poorly designed circuits, buildings, paintings, music, etc.

Abstraction teleology relates the behavior of the system's components to its ultimate purpose by summarizing the behaviors of groups of constituents until satisfaction of the purposes has

been verified. Electrical engineering has a standard taxonomy of combinations of configurations. Associated with each configuration is the purposes it can meet. QUAL utilizes a *plan library* of such circuit mechanisms to determine the abstraction teleology of the overall circuit. In this way QUAL can identify the highest level concepts that engineer's use.

The causal analysis provides the semantics for the teleological calculus. Envisioning is the basis for implementation teleology, which in turn is the basis for abstraction teleology. Envisioning deals with only a small set of the characteristics that a circuit module can have. However, the few properties it does identify can be used to index into the library to determine what other properties the module must have. For example, one does not have to determine that an engine uses fuel to recognize it as an engine — mechanical clues are sufficient. However, all engines consume fuel, therefore the device in question consumes fuel although you may not have noticed it. Furthermore, you can tell whether the engine uses oil or gasoline by looking at the ignition system. In electronics, the envisioning provides a sufficiently complete set of features such that all of the other characteristics of the module can be identified. The plan library is, in effect, a calculus by which the characteristics of the constituent components can be summarized and related to the systems ultimate purpose.

Chapter 4

CAUSAL REASONING

4.1 Causal Explanations of Circuit Behavior

This chapter presents a theory of causal analysis. Instead of immediately describing the causal analysis process and the associated device models, this chapter is organized starting with obvious but inadequate notions causal reasoning and developing them when they fail to explain some circuit's behavior. In this way the reader can gain some appreciation for the underlying motivations for the structure of the theory.

When an electrical engineer is asked to explain the operation of an electrical system he will often describe it in terms of a sequence of events each of which is "caused" by previous events. Each event is an assertion about some behavioral parameter of some constituent of the system (e.g. current through a resistor). By throwing away most of the details of the system, he is able to extract a sequential description of the behavior of the system, characterizing its major features. This crude characterization of circuit behavior is sufficient for many purposes.

Sequential descriptions are ubiquitous in engineers' verbal and textbook explanations. Consider the Schmitt trigger (see figure 1). The explanation reads as if a time flow has been imposed on it.

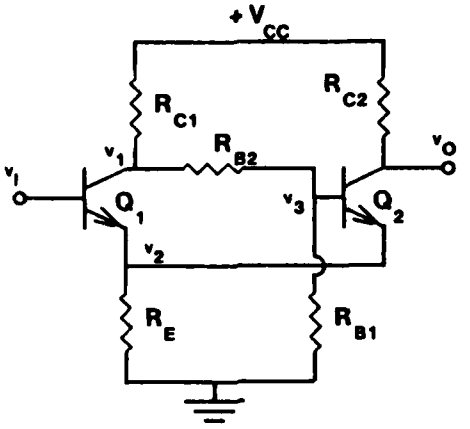


Figure 1 : The Schmitt Trigger

" ... An increase in v_I augments the forward bias on the emitter junction of the first transistor, thereby causing an incremental increase in the collector current, i_{C1} of that transistor. Consequently both the collector-to-ground voltage v_1 of the first transistor, and the base-to-ground voltage of the second transistor v_3, decrease. The second transistor operates as an emitter follower which has an additional load resistor on the collector. Therefore, there is a decrease in the emitter-to-ground voltage v_2. This decrease in v_2 causes the forward bias at the emitter of the first transistor to increase even more than would occur as a consequence of the initial increase in v_I alone ..." [Harris *et.al.* 66, p.68]

A goal of this research is to develop a clear understanding of the notion of causality as found in this argument.

Causal explanations describe how the behaviors of individual components contribute to the overall behavior of the system. This knowledge is important for understanding, designing and troubleshooting designed systems. A complete algebraic analysis of even simple circuits can be computationally prohibitive, but knowledge of how the individual components contribute to the circuit's composite behavior can significantly improve the efficiency of the analysis [de Kleer & Sussman 78]. For example, an integrated circuit operational amplifier contains a large number of transistors, but few of them are situated on the main signal path. For many calculations the effect of these auxiliary transistors on the signal can be ignored or accounted for by much simpler transistor models. The causal explanation identifies which transistors are crucial to the behavior and which are not. The use of these simpler models significantly reduces the complexity of the algebraic analysis.

Causal reasoning also plays a fundamental role in identifying the faults responsible for symptomatic behavior and in localizing faults at a shallower level of detail before entering the more expensive deep analysis [Brown 76] [de Kleer 76]. Early designs can be checked to see whether they have any hope of achieving their desired behavior, and the sections which are critical to the desired behavior can be identified for special attention [McDermott 76].

4.2 Causality is an Artifact

The "causality" of an argument is an artifact of the level of detail used in the analysis that produced it. This can be demonstrated in the Schmitt trigger example by using a transistor model whose v_{BE} is fixed. Using this model v_3 still drops as a consequence of increasing i_{C1}, but v_2 now rises since v_I is rising and v_{BE} is fixed. Both of these effects cause v_O to rise. This new argument predicts the same output signal, but the details of how this signal is achieved are completely different. The new argument does not identify the feedback, and predicts that v_2 will rise while the earlier argument predicts it will drop. This is an example of two different causal explanations for the external behavior.

Since the component models utilized in these causal arguments are local, these arguments could all have been generated by a simple propagation of known signals: the signals are applied to their adjacent device models which in turn predict other signals. Although most causal arguments can be generated by propagation, no such claim can be made about the validity of the converse. With a rule "A causes B," propagation will deduce B if A is valid, but will also deduce A if B is valid when there is no other plausible cause to account for it. The latter deduction is often undesirable. For example, one usually thinks of increased v_{BE} causing increased i_C, but the inverse deduction of increased i_C causing increased v_{BE} is usually thought to be noncausal since something must have increased v_{BE}. However, the collector current cannot increase unless something supplied it with more current. This example further illustrates that "causality" is largely an artifact of the point of view taken to analyze the circuit.

The explanation for the Schmitt trigger made a number of unsubstantiated assumptions aside from the choice of transistor models. Why does the v_I increment appear across Q_1 instead of R_E? Why does the voltage v_1 drop since Q_2's turning off should raise it? Why is the current contributed by Q_2's turning off more than the current taken by Q_1's turning on? There are many values for the parameters for which the circuit cannot function at all. The arguments are

only rationalizations of the observed behavior (observed by actual measurements or stated in the textbook). This does not detract from the usefulness of the explanations: no explanation ever accounts for every detail of the behavior. The usefulness of an explanation does not depend on how complete or correct it is, but whether the explanation is sufficient for the purposes it is applied to.

4 3 The Machinery for Causal Analysis

This section develops a mechanistic model for causal reasoning. The purpose of the model is to explain how causal arguments can be discovered. A causal argument consists of a sequence of assertions about electrical quantities each of which hold as a consequence of previous assertions. For example, the causal argument " ... An increase in v_I augments the forward bias on the emitter junction of the first transistor, thereby causing an incremental increase in the collector current, ... ," is a sequence of two assertions: v_I increases, i_{C1} increases. These assertions are the *events* of a causal argument. The deduction of one event from another is determined by device models. In the above example the model for the first transistor is one in which increased emitter potential causes increased collector current. The device models are central to the theory since they utilize a description of the topological structure of the circuit to specify the rules underlying the behavior of the circuit. These models are the only part of the theory that refers to circuit topology; all further theory will utilize the mechanism fragments that the models produce.

Before discussing the formal computational machinery for causal analysis, let me present some of the underlying intuitions. Causal analysis produces a causal argument which is a qualitative description of how the circuit equilibrates – how it responds to perturbations from its equilibrium state. This description is, in effect, a simulation of the circuit's equilibration. This kind of explanation is often what people mean by a description of how some thing "works." Not surprisingly, the causal analysis process itself is also a simulation. It, in effect, simulates many possible causal arguments simultaneously but uses a variety of strategies and heuristics to evade the potential combinatorial explosion. The remainder of this section presents the formal machinery suggested by these intuitions and may be skipped on first reading.

The causal analysis machine is based on the presuppositions that the causal device rules are local and that the events of a causal argument are discovered in their causal sequence. By local I mean that the rules for a device (1) refer to a small number of circuit quantities, (2) refer to

circuit quantities that are topologically adjacent to the device being modeled, and (3) that every device of the same type is modeled by the same rules, independent of topological context.

The causal analysis machine has three components. The modeling component specifies the behavior of the basic devices. The wiring component provides a way to connect the basic devices to describe circuits and composite circuit models. The execution component determines when device rules are to be applied. The wiring and execution components are almost completely determined by the presuppositions of locality and ordering. The modeling component will be discussed after the wiring and execution have been developed.

The quantities of interest in the analysis of a circuit are represented by *cells*. Each voltage, current and device parameter has its own unique cell. A cell may contain one or more values. For example, i_{C1} may be represented by CELL-67 and contain the values 1 ma and 0 ma signifying that the collector current is 1 ma when Q1 is on and 0 ma when Q1 is off. Each cell is connected to the other cells by electrical laws. Whenever a cell receives a new value the rules it participates in are examined to determine whether it is possible to deduce new values for neighboring cells. Since a cell can participate in many rules, a queue of newly discovered values is maintained. This can introduce nondeterminism. If only one value can be deduced from each application of a rule, the queue will not grow and the assignments will be totally ordered.

The behavior of an electrical component is described by a device model which consists of an association list specifying the cells the model is connected to and rule prototypes referring to these cells. The rule prototypes specify how values in the cells are related. When a new circuit is created, instances of the circuit models are created for each of the circuit's devices. An instance of a model is constructed by making a copy of the rule prototype and connecting it to the transistor's cells as indicated by the association list. The wiring component provides a very general mechanism. SYN [de Kleer & Sussman 78] uses the same machinery to do synthesis of electronic circuits by propagation of constraints (see section 2.5 for a discussion of SYN). In the case of propagation of constraints the rule prototypes are algebraic equations. Although the structure of the device models and how they are used to construct a composite model for the entire circuit raise many important and difficult issues, these issues are not central to this chapter which is primarily concerned with the content, not the structure of the models. Therefore the models described in this chapter will accurately reflect only their content. Appendix 1 presents the more complete LISP-based formulations of the models.

Causal flow analysis, which describes circuit behavior in terms of a sequence of events, is distinguished from other types of analysis by how it deals with time. Causal analysis assumes that the time of the basic machine can be identified with the sequential events of the causal argument, later events in the argument are discovered later in the analysis. Each event in a causal argument is an assignment of a value to a cell. This value depends on previous events in the argument, and *must not be changed or improved upon* after it has been placed there. Some of the consequences of this are that each cell is assigned a value only once and that each rule is used only unilaterally. A rule is used *unilaterally* if each of the cells it is connected to is used only as an output or as an input, but not both. If a rule uses the same cell as an input or output, it is used *bilaterally*. Analogously a rule which has the potential to be used bilaterally is referred to as bilateral rule.

Propagation of constraints violates most of these conditions when it introduces anonymous objects. Cell values which depend on anonymous objects change as the system solves for the anonymous objects. In order for propagation of constraints to solve for the anonymous objects, the rules must be expressed as bilateral constraints. The rules used in causal models, however, tend to be unilateral: transistor v_{BE} can cause i_C, but not vice versa. The conditions of causal flow analysis demand that every bilateral rule be used only unilaterally. For example, the causal resistor model is bilateral in i_R and v_R, but the rule must be used only unilaterally in a particular causal flow argument: for any resistor, i_R must be used to derive v_R, or vice versa, but not both.

An analysis by propagation of constraints that does not require the introduction of anonymous objects meets the criteria for a causal flow analysis, but such analyses are rare. A causal flow analysis is permitted to make assumptions about the behavior of the circuit. Assumptions, like anonymous objects, are used to break impasses in the analysis. However, an assumption is not a kind of disguised anonymous object. The anonymous object is introduced in the hope that the ensuing propagations will be able to restrict the anonymous object's value. A propagation based on an assumption has a completely determined value and this value does not change if the assumption is validated or refuted. (If the value is refuted, it just ceases to be of interest to the analysis.) Assumptions provide a way of expressing partial information about the circuit's behavior. A value which depends on an anonymous object is unknown, but a value which depends on an assumption is known if the assumption is valid. The applicability of causal flow analysis depends on how easy it is to compute with these assumptions. Although it is easy to express assumptions

represented as algebraic expressions, it is difficult to compute with them.

The machine can be controlled in two distinct ways. The queue of pending deductions can be reordered arbitrarily and the rules upon which it operates are arbitrary. These two techniques allow the machine to be controlled such that the implicit time order of its deductions is identified with the time imposed by a causal argument. The basic idea is that device models are forced to be *locally causal*. For example, the causal model for the transistor does not respond to changes in i_C. Deductions based on assumptions are inserted at the end of the queue.

4.4 Electrical Device Models

The classical engineering models that are used to describe the behavior of electrical components are widely agreed upon. However, the causal qualitative models that people use to reason about circuits are not. In fact, these qualitative models are rarely articulated, even though the tacit models that underlie people's arguments appear to be very similar. This section presents a sequence of different models for a few devices in order to explain the issues involved. A simple model will be proposed first, followed by more sophisticated models designed to correct the shortcomings of the first.

Although a circuit quantity can be represented by a single *total* variable, engineers usually consider it as the sum of an *incremental* component and a *quiescent* component. The quiescent component represents the value of the circuit quantity when no signal is present and the circuit has reached steady state. The incremental component represents the deviations from this quiescent value which occur when signals are applied.

The causal explanation of how a circuit works is a qualitative description of the equilibrating process that ensues when signals are applied to the circuit. The behavior of the Schmitt trigger was described in this way. This will be called *incremental qualitative* (IQ) analysis. Since most circuits are designed to deal with changing input signals, it is not surprising that the main purpose of most circuits is achieved incrementally. For example, an amplifier must amplify changes in its input, digital circuits must switch their internal states as applied signals change, and power-supplies must provide constant current or voltage in the face of changing loads and power sources. For these kinds of circuits, the purpose of the quiescent behavior is to put the nonlinear devices into a desired region of incremental behavior. Since quiescent behavior plays a subsidiary role in the incremental analysis, this thesis concentrates on the latter.

Incremental qualitative arguments rarely need to refer to more than the sign of the derivative which indicates whether the signal is increasing or decreasing. This requires an algebra of four values: "⇑" signal is increasing, "0" signal is not changing, "⇓" signal is decreasing, and "?" signal is unknown. The arithmetic of this algebra is very simple:

	x:	⇓	0	⇑	?
y:					
	⇓	⇓	⇓	?	?
	0	⇓	0	⇑	?
	⇑	?	⇑	⇑	?
	?	?	?	?	?

Table 1 : x + y

Only addition and subtraction are important, and no other operations are ever used. Anonymous objects are never used in causal arguments and are thus unnecessary. These restrictions make the algebra subsystem of the machine trivial.

The simplicity of this algebra is deceptive. Note that the IQ value represents the change of the value of the quantity, and not the change of the magnitude (absolute value) of the quantity. For example, if x changes in value from -6 to -7 its value is decreasing, even though its magnitude is increasing from 6 to 7. Thus, the statements "x is increasing" and "$-x$ is decreasing" are equivalent. There are many possible alternate algebras, two of which are worth analyzing since they appear to be more plausible than the one chosen here. The IQ value could represent the change of the magnitude of the quantity. In this case a change from -6 to -7 is considered an increase. One problem with this algebra is that addition is no longer unique. If x is increasing and y is increasing, $x + y$ could be decreasing. For example, if x increased from 3 to 4 and y increased from -1 to -3, $x + y$ would decrease from 2 to 1. Although this algebra is blatantly inadequate it has been my experience that this is the algebra of choice with beginning students. The second alternate algebra is a modification of the previous one to include the sign of the quantity as well as its change. From the sign of a quantity and the sign of the change in absolute value of a quantity it is possible to determine the sign of the change of the quantity itself. This algebra is unnecessarily complicated since requires knowledge of the signs of

all the quantities while the algebra of table 1 does not. Section 9 will introduce some conventions for explanation which include some of the desirable features of the these two alternate algebras without their undesirable problems.

The approach for constructing the models is to start with the classical constraint models, and reformulate them preserving only the sign of the derivatives of the variables. Ohm's law has a particularly simple formulation. The conventions are that voltage is defined with respect to two nodes, currents are defined as flowing in terminals (the paths connecting devices to nodes), and the sign convention is chosen such that current flowing into devices away from nodes is positive. This sign convention for currents is rather clumsy for causal arguments and a far more intuitive one will be adopted in section 9 after causal reasoning has been analyzed in detail.

$v = iR$ $\qquad\qquad$ $\updownarrow v = \updownarrow i$

Figure 2 : Ohm's Law

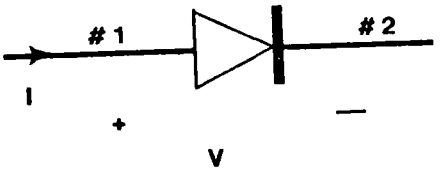
$\updownarrow x$ refers to the sign of the derivative of x. The IQ model for Ohm's law is:

$$\boxed{\updownarrow v \Leftrightarrow \updownarrow i}$$

Model 1

The definition of the variables which appear in the models is given by the component's diagrams. Figure 2 defines i to be the current flowing into the positive terminal, and v to be the voltage between the positive terminal and the negative terminal. (Remember that the convention is that current flows into devices away from nodes.) Kirchoff's Current Law (KCL) automatically applies to components so that the current in the top terminal is equal and opposite to the current through the bottom terminal. The model specifies that the derivative of the current must be of the same sign as the derivative of the voltage. Since the resistor has no preferred causal flow direction this rule must be bilateral. This action is specified by the "$\Leftrightarrow$" operator.

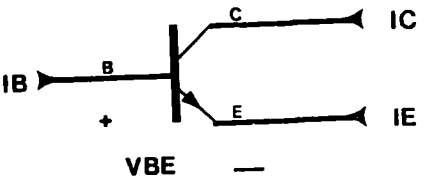
The ideal diode conducts zero current when the voltage across it is below a certain threshold and conducts an arbitrary amount of current at that threshold. This behavior is usually modeled by the two states on and off:




```
if D is on, ↕v = 0
if D is off, ↕i = 0
```

Model 2

In the off state, the current through the diode is zero as well as all of its derivatives. The above model, however, only indicates that the current is unchanging (i.e. the first derivative is zero). A particularly simple model for a transistor has an ideal diode as its emitter junction and a controlled current source at its collector:




```
if Q is on, ↕vbe = 0, ↕ib ⇒ ↕ic, ↕ib -⇒ ↕ie
if Q is off, ↕ib = 0, ↕ic = 0, ↕ie = 0
if Q is sat, ↕vbe = 0, ↕ic = 0
```

Model 3

The "⇒" operator is like ⇔ except that it specifies an assignment in one direction only. Note that ⇒ and ⇔ always refer to derivatives. The "-⇒" operator behaves like ⇒ except that it inverts the sign of the quantity.

4.5 Analysis of a DTL-Inverter

In order to analyze a circuit containing devices which have different states, the various composite circuit states must be considered. Sometimes the applied signal can force a unique state choice, and sometimes a number of possible circuit states have to be explored simultaneously. Transistor and diode models assert values which are dependent only upon the state they are in. A *state-value* assertion for a transistor in the off state is `↕ib = 0`. State-value assertions can be invoked without propagations, but in order to prevent a proliferation of circuit states, the state-values of a model are only used if a signal is detected near the device. In this way new circuit states will only be considered when necessitated by the propagation. A signal can also cause the circuit to change state. The rules for such state transitions will be discussed later.

These are enough device models to analyze the simplified DTL (Diode-Transistor-Logic) inverter which is constructed from transistors, diodes and resistors:

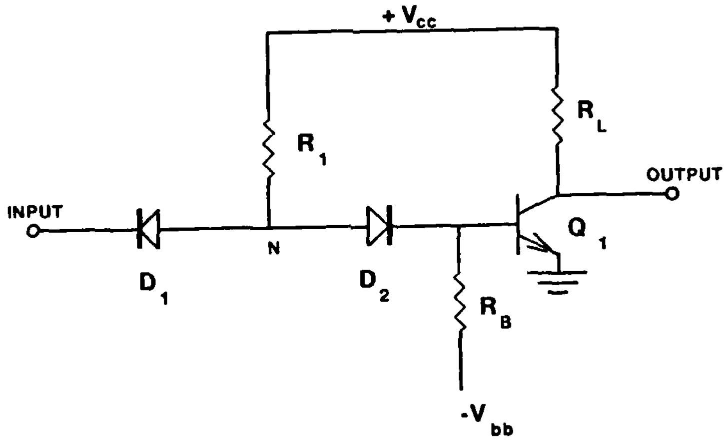


Figure 3 : DTL-Inverter

When a voltage signal is applied to the input, nothing happens since the diode model only operates on current through the diode, or voltage across it. Since the simple diode model only has outputs, the analysis must make an arbitrary choice as to whether D1 is on or off. If D1 is off, the current through it is zero and propagation halts indicating that the rest of the circuit values remain unchanged. If D1 is on, an increase in input voltage results in an increase at N. A similar analysis applies to D2. If D2 is off, the remaining circuit values are unchanged. If D2 is on, the voltage at the base of Q1 is rising which is only possible if Q is off. If Q1 is on, the

model says that its base-emitter voltage cannot vary. Thus the models are inadequate to analyze the DTL-inverter.

The DTL analysis failed to explain how the inputs to the circuit affect its output. The ideal diode model for Q1 produces a contradiction when Q1 is on. Even ignoring the contradiction, the models for D1 and D2 do not say anything about the current flowing through them. Therefore no signal would appear at i_B or the output node. One possible solution is to include the exponential diode effect for Q1.

```
If Q is on, ↕vbe ⇒ ↕ib, ↕vbe ⇒ ↕ic, ↕vbe -⇒ ↕ie
```

Model 4

Instead of

```
if Q is on, ↕vbe = 0, ↕ib ⇒ ↕ic, ↕ib -⇒ ↕ie
```

Model 5

This evades the contradiction. Unfortunately, if this exponential diode model is used for D1, the analysis can no longer determine whether the voltage at N drops. Furthermore, D1 and D2 could have their polarities reversed without affecting the analysis:

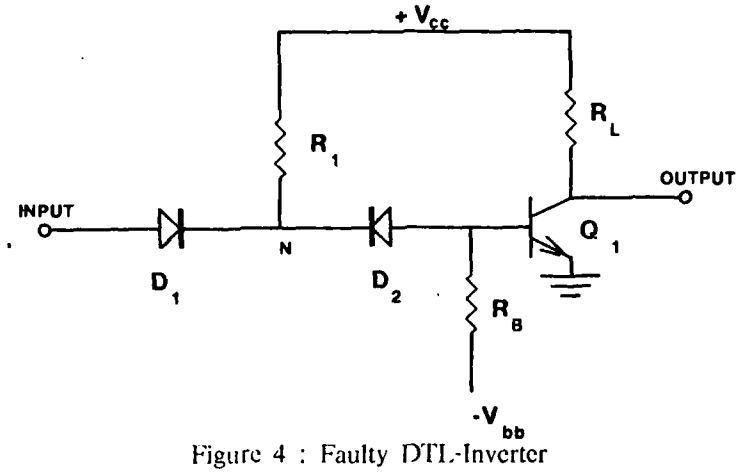
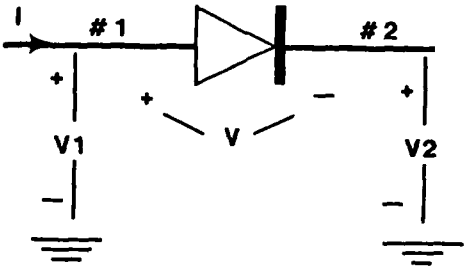


Figure 4 : Faulty DTL-Inverter

In the correct circuit the current through D1 decreases as the input signal rises. In the faulty circuit this current increases. Since no external voltage is discovered across D1, the exponential diode model cannot be utilized to determine the direction of current flow.

Consider a causal argument a person might give for the inverter's operation: "As the input signal rises, N rises and the current through D1 drops. As N rises, D2 turns on harder, increasing the current through it and raising the base of Q1. Q1 turns on harder and pulls down the output." Note that each device appears only once and its model is often invoked upon insufficient evidence. For example, D1 can only communicate the signal to N if the voltage at N is higher if D1 is removed. The current through D1 decreases only if N does not rise faster than the input. The model employed to describe D1 makes the presupposition that this is case. Stated differently, the diode model always makes the presupposition that the first signal detected near the diode invokes the model as if this signal dominates all of the other quantities the model references. In this simple circuit these presuppositions can be trivially verified, but there is no way the diode model, which only has access to local information, can determine this.

This is the beginning of the notion of a causal argument. To reiterate, a causal argument consists of a sequence of events, each event describing how the behavior of a node or device is influenced by earlier events, with the presupposition that the earliest discovered trigger signal is the dominant input to that node or device. Assuming that the ordering of the events within the execution component can be identified with the sequence of the causal argument, the *causal presupposition* can be incorporated into the models. The diode model now becomes:



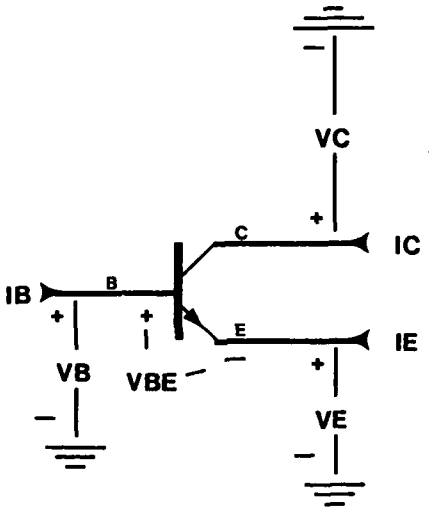

```
if D is on, ↕v ⇒ ↕i
            ↕v1 C⇒ ↕v2, ↕v1 C⇒ ↕i, ↕v1 C⇒ ↕v
            ↕v2 C⇒ ↕v1, ↕v2 C⇒ ↕i, ↕v2 C-⇒ ↕v

if D is off, ↕i = 0
```

Model 6

The "C⇒" operator acts like ⇒, except that it acts only on nonzero values. In order to include

the consequences of the causal presupposition explicitly, models refer to voltages at their terminals as well as voltages across their terminals. The causal presupposition assumes all values are zero, so it is never necessary to propagate zero values. In fact, a zero input should never be considered a dominant input, even if it is found first. Not propagating these zero quantities cannot result in erroneous analyses since in those cases where the zero value would have participated in a contradiction, the value it would have contradicted with must be nonzero and that value will have propagated causing a contradiction at a different place.

The model for a transistor now becomes:




```
If Q is on, ↕vbe ⇒ ↕ic, ↕vbe -⇒ ↕ie, ↕vbe ⇒ ↕ib
            ↕vb C⇒ ↕ve, ↕vb C⇒ ↕vbe, ↕vb C⇒ ↕ib
            ↕vb C-⇒ ↕ie, ↕vb C⇒ ↕ic
            ↕ve C⇒ ↕vb, ↕ve C-⇒ ↕vbe, ↕ve C-⇒ ↕ib
            ↕ve C⇒ ↕ie, ↕ve C-⇒ ↕ic

If Q is off, ↕ib = 0, ↕ic = 0, ↕ie = 0

If Q is sat, ↕ic = 0
```

Model 7

If the transistor is directly connected to the local reference, ve and vb are not utilized. This is

the case with the transistor in the DTL-inverter.

Employing these models the DTL analysis succeeds. The following is the causal argument that QUAL finds for the output behavior. The format of this explanation is a causally-ordered sequence of events described by cell-value pairs, each of which is followed by a one line explanation of the model rule that deduced it. Since events can have multiple antecedents and consequents, only simple causal arguments can be expressed with a totally ordered linear list. When an event has multiple consequents or antecedents this fact will be indicated in the causal argument and the argument for that value will be included in a judicious place in the event sequence. In general, there are many events caused by the inputs which do not affect circuit outputs. These will not be included in the causal arguments. (Note that V2 is the voltage on the left terminal of D1.)

```
Starting with input:
(VOLTAGE INPUT GROUND) = ⇑
   Premise.
(VOLTAGE N GROUND) = ⇑
   V2 C⇒ V1 for D1
(VOLTAGE B GROUND) = ⇑
   V1 C⇒ V2 for D2
(CURRENT C Q1) = ⇑
   V ⇒ IC for Q1
(CURRENT #2 RL) = ⇓
   KCL for node OUTPUT
(CURRENT #1 RL) = ⇑
   KCL for device RL
(VOLTAGE OUTPUT VCC) = ⇓
   ⇔ V I for RL
Also given that:
(VOLTAGE GROUND VCC) = 0
   POSITIVE-SUPPLY
The combination of events (VOLTAGE VCC GROUND) (VOLTAGE VCC OUTPUT) cause:
(VOLTAGE OUTPUT GROUND) = ⇓
   KVL applied to nodes GROUND VCC OUTPUT
```

The deductions the models make depend upon the order in which the propagator discovers

new values. Suppose a rising voltage is applied to a transistor. If the increase is applied to the base, the emitter must follow and the collector current increases. If the increase is applied to the emitter, the base will rise and the collector current decreases. Taking into account only the voltages at the base and emitter, the two examples are identical. The collector current is determined by which voltage the propagator found first. The causal presupposition says that the collector current is determined by that voltage which caused the other.

Figure 5 : Causality at the Emitter Junction

The causal presupposition can be violated, and the propagator must detect these violations. Whenever a model makes a deduction based on the presupposition it should explicitly mention which values are assumed to be negligible with respect to triggering quantity. If this assumption is ever violated, the propagator should retract the original deduction. Causal presuppositions can also make subsequent teleological reasoning more difficult since the sole purpose of a circuit fragment may be to ensure the nondominance of a quantity. If a causal presupposition is made that this quantity is nondominant, the purpose of the circuit fragment cannot be determined. To avoid this difficulty, the propagator should try to substantiate all of its causal presuppositions after the analysis is completed.

4.6 The KVL Connection Heuristic

The rules of the device models are of two different types: rules which involve assumptions that do not necessarily hold, and basic rules which involve no assumptions and are universally valid. The $\updownarrow v_{BE} \Rightarrow \updownarrow i_C$ transistor rule makes no assumptions and is thus a basic rule. An example of a heuristic rule is $\updownarrow v_B \Rightarrow \updownarrow i_C$ which assumes that the v_B input is dominant. In order to reason about and possibly retract these assumptions, the assumptions themselves have to be explicitly recorded.

The heuristic rule $\updownarrow v_B \Rightarrow \updownarrow i_C$ makes an assumption about the behavior of the circuit around the transistor and not about the transistor itself. If this heuristic voltage rule is consistently applied

to all the device models, every basic voltage-difference rule must be expanded into two separate heuristic voltage rules. These voltage rules specify how the individual device models are connected together, and therefore a special *KVL connection heuristic* is introduced to replace them. The KVL-heuristic is implemented as a procedure which is triggered whenever a nonzero voltage is discovered at a node. It looks for device models with voltage inputs that refer to this node and triggers them. For example, when the KVL-heuristic discovers a voltage at the base of a transistor, it triggers the model on its base-emitter voltage. In doing so, the rule makes the assumption that the emitter voltage's effect is negligible compared to that of the base voltage. The assumption that the base voltage is the dominant input to Q is recorded as $[Q\ v_B]$. Under this assumption a rising base voltage will thus cause a rising collector current.

If the voltage at the emitter is discovered to be rising independently, the KVL-heuristic determines that the collector current is falling under the assumption $[Q\ v_E]$.

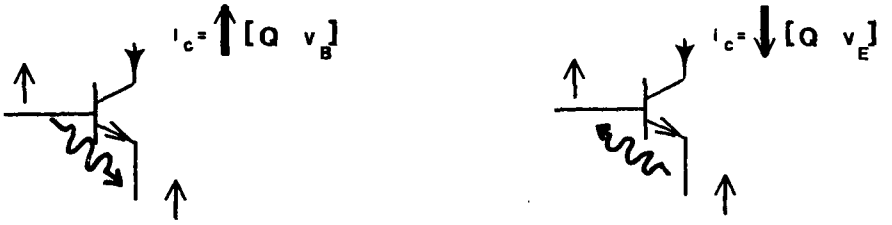


Figure 6 : v_B and v_E Assumptions

Since the two contradicting values for i_C hold under different assumptions, the only effect of the contradiction is to record that at least one of the assumptions is invalid. The introduction of explicit assumptions has freed the analysis process from the nondeterminism introduced by the queue; no matter when v_B is discovered, it will propagate to i_C since that propagation step involves a new independent assumption different from any other assumptions that were made about that transistor.

KVL is inherently a constraint law. One possible causal implementation of this constraint attempts all possible consistent assignments of values to the individual branch voltages. If the quiescent current flow directions are known, the situation is improved but the strategy generates far too many assumptions to be useful. Instead, the KVL-heuristic assigns a value only to the outermost branch voltage, other rules being expected to propagate this voltage to the individual

branches. The input voltage of the Schmitt trigger appears across the input transistor and emitter resistor. Since the transistor is connected to the input voltage, it is the device that receives the input voltage rather than the resistor which is connected to the neutral reference. The voltage across the resistor must be calculated by the other rules.

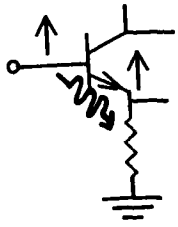


Figure 7 : Schmitt Trigger Input

From an equilibrium point of view this KVL-heuristic is false, but it captures the kind of causality manifested in the Schmitt trigger explanation. The KVL-heuristic also makes the presupposition that all interesting voltages eventually propagate to a voltage with respect to a common reference. This presupposition is false in analog multipliers and other heuristics have to be developed to deal with such circuits. These can be analyzed if more references are introduced, but this results in excessive redundant arguments as well as requiring *a priori* knowledge of circuit behavior.

Associated with each propagated value is an *environment* descriptor which indicates the circuit state it applies to and the assumptions under which it is valid. If incompatible environments are kept separate, different environments can be explored simultaneously. In this way those areas of circuit behavior which are common among environments can be shared. Two environments are incompatible if one environment contains an assumption or state choice on a device and the other environment contains a different assumption or state choice on this same device. Thus any environment which contains [Q v_B] is incompatible with any environment which contains [Q v_E].

The IQ model for a transistor is represented as follows:

```
If Q is on, ↕vbe ⇒ ↕ic, ↕vbe ⇒ ↕ie, ↕vbe ⇒ ↕ib
If Q is off, ↕ib = 0, ↕ic = 0, ↕ie = 0
If Q is sat, ↕ic = 0
```

Model 8

As the KVL-heuristic applies universally when a voltage is discovered at a node, any causal input voltage which refers to that node (and some other) is assumed to also receive this voltage value. Note that both NPN and PNP transistors have the same IQ model.

The rules of the device models come from the algebraic models used in electrical engineering and from the rough qualitative models observed in engineers' arguments. Since there is a diversity of algebraic and qualitative models, there is also a variety of incremental qualitative models. The standard algebraic incremental and quiescent models (Hybrid-π and Ebers-Moll) for a transistor employ a dependent current source to describe the collector current. A current source only constrains current and not voltage, therefore the IQ transistor model describes the collector current as a causal output and ignores the collector voltage. The causal action of the emitter junction is more complex, and the IQ model is based on the observed arguments engineers use. A simple model has $\updownarrow v_{BE} = 0$ and i_B as a causal input. In most situations the diode behavior is necessary to explain i_C (although $\updownarrow i_B = 0$, the infinite-beta model holds more generally). Although the exponential diode equation does not distinguish between voltage and current, the diode action is almost invariably described as a voltage causing a current, as seen in the fact that the diode equation is always written as an exponential. Mathematically, a logarithmic equation is just as accurate. Therefore the basic IQ transistor model treats v_{BE} as a causal input and i_B (if beta is finite) and i_C as outputs. The only rule an IQ model must obey is that it assert all the voltages and currents associated with the device, because a device model cannot trigger on its own outputs (or any consequence thereof).

An examination of electronics textbooks shows the dominance of voltage as a causal quantity. For example, voltage is explained as a force and current as the stuff moved by this force. The various IQ models and heuristics follow this convention. Mathematically there is no reason to distinguish between voltage and current and there are some circuits which are better understood in terms of currents, but they are relatively rare and will not be discussed here.

4.7 The KCL Connection Heuristic

The exponential diode model used in the DTL-inverter analysis has a rule which states that the voltage on the anode follows the voltage on the cathode. This rule makes assumptions about the behavior of the rest of the circuit. In particular, the rule assumes that the diode is not connected to a negative resistance. There are many other situations in which it is useful to make

this kind of an assumption. The emitter junction of a transistor behaves as an exponential diode in that the voltage on the emitter usually follows the voltage on the base. Increased collector current usually pulls down the voltage on the collector node.

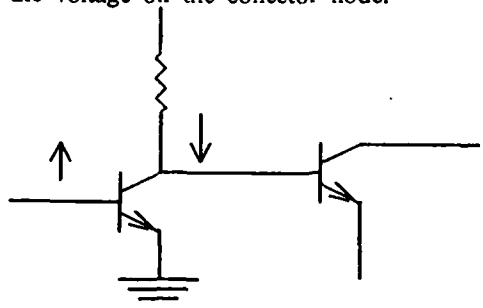


Figure 8 : Collector Current Pulling down Node Voltage

If the current through a resistor is caused to drop, the voltage at its positive terminal usually drops as well. In all these situations, current flow into a node affects voltage at the node. One way to make these kinds of deductions possible is to follow the example of the simple diode model and add heuristic rules to every model with causal current outputs.

This unnecessarily complicates the device models and requires even the basic models to make assumptions. Like the KVL-heuristic, this heuristic is really a statement about the behavior of the rest of the circuit, and not about the particular device causing the changing current. For these reasons a separate node model is used which models the behavior of nodes.

The *KCL-heuristic* is implemented by a procedure. If the node voltage is unknown, and some of the currents into the node are known, then the voltage at the node rises if the sum of the currents ignoring KCL on the node is positive, and drops if the sum is negative. This assumption is recorded as [<node> <terminal1> ... <terminaln>]. The KCL-heuristic must be applied to every environment individually since a voltage known in one environment can be unknown in another. Since the KCL-heuristic makes such a major assumption about circuit behavior, and since it can be more judiciously applied if more currents and voltages are known, it is run after all possible propagations have been made in the environment onto which it will assert the new node voltage.

From the point of view of network theory, the KCL-heuristics make the assumption that the terminals which are causing current flow into the node can be modeled as the terminal of a current source, and that the remaining terminals can be modeled as the terminal of a (incrementally) positive resistance:

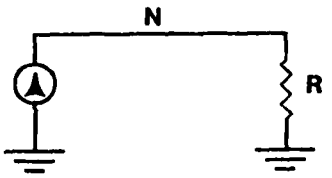


Figure 9 : KCL-Heuristic Network Assumption

The KCL-heuristic assumption can be violated. This is especially true in circuits with feedback. The KCL-heuristic can also be redundant in that the voltage at the node is either irrelevant or can be deduced in some other way.

The existence of the KCL- and KVL-heuristics make the IQ diode model very simple:

```
If D is on, ↕v ⇒ ↕i
If D is off, ↕i = 0
```

Model 9

4.8 Example IQ Analyses

In summary, the basic IQ machine employs three kinds of rules: model rules, KCL and KVL, and KCL- and KVL-heuristics. The rules of the device models are locally causal and do not make assumptions. KVL and KCL apply when all but one of a collection of currents or voltages is unknown. They also make no assumptions. Finally, the KCL- and KVL-heuristics allow the analysis to connect together the behavior of the local device models. Since KCL and KVL are inherently constraint-like, these two heuristics introduce an artificial equilibration time by making their assumptions explicit.

One purpose of the assumptions is to free the IQ analysis from the nondeterminacy of the queue of the basic propagation machine. If every possibly invalid event makes an explicit assumption, the order in which the events are found will have no effect on the ensuing contradictions. A second equally important purpose of assumptions is to identify the cause of a contradiction. The IQ rules may make far more assumptions than necessary. For example, no assumptions are logically necessary in the causal analysis of the DTL-inverter because all the IQ

rules are completely local. In the case of the KCL-heuristic at node N, it was unable to tell that there was not a feedforward path to the top of R1. A rudimentary topological analysis could have determined this, but the KCL-heuristic cannot do any topological analysis and therefore must be prepared for the worst. There is however, a simple strategy to remove many of the assumptions. At the conclusion of the analysis all the possible causes have been investigated, and therefore any assumption that does not immediately lead to multiple values must hold. By this strategy, all the assumptions made in the DTL analysis are verified.

The following is the causal argument for the DTL-inverter utilizing the new models: resistor model 1, diode model 9, and transistor model 8. Each event is followed by the list of assumptions (the environment) made by the causal argument that far.

```
Starting with input:
(VOLTAGE INPUT GROUND) = ⇑ <>
   Premise.
(VOLTAGE N INPUT) = ⇓ <[D1 V2] (D1 ON)>
   KVL-heuristic [D1 V2]
(CURRENT #1 D1) = ⇓ <[D1 V2] (D1 ON)>
   V ⇒ I for D1
(VOLTAGE N GROUND) = ⇑ <[N D1] [D1 V2] (D1 ON)>
   KCL-heuristic [N D1]
(VOLTAGE B N) = ⇓ <[D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KVL-heuristic [D2 V1]
(CURRENT #1 D2) = ⇑ <[D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   V ⇒ I for D2
(CURRENT #2 D2) = ⇓ <[D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KCL for device D2
(VOLTAGE B GROUND) = ⇑ <[B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KCL-heuristic [B D2]
(CURRENT C Q1) = ⇑
      <(Q1 ON) [B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   V ⇒ IC for Q1
(CURRENT #2 RL) = ⇓
      <(Q1 ON) [B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KCL for node OUTPUT
```

```
(CURRENT #1 RL) = ⇑
      <(Q1 ON) [B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KCL for device RL
(VOLTAGE OUTPUT VCC) = ⇓
      <(Q1 ON) [B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   ⇔ V I for RL
Also given that:
(VOLTAGE GROUND VCC) = 0 <>
   POSITIVE-SUPPLY
The combination of events (VOLTAGE VCC GROUND) (VOLTAGE VCC OUTPUT)
      cause:
(VOLTAGE OUTPUT GROUND) = ⇓
      <(Q1 ON) [B D2] [D2 V1] (D2 ON) [N D1] [D1 V2] (D1 ON)>
   KVL applied to nodes GROUND VCC OUTPUT
```

Since no conflicting multiple values are found, the assumptions <[B D2] [N D1] [D2 V1] [D1 V2]> are verified.

The analysis of the emitter-coupled pair further illustrates the use of the connection heuristics.

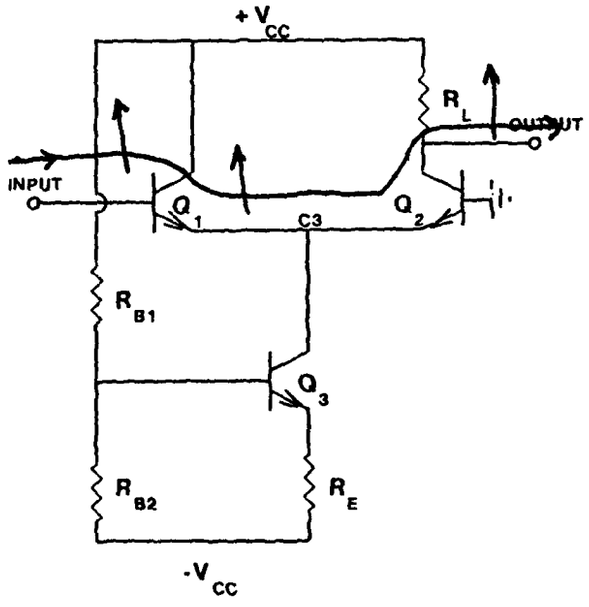


Figure 10 : Emitter-Coupled Pair

The light line drawn through figure 10 indicates the main signal path, and the vertical arrows on that path indicate changes in the potential of the nearby nodes.

```
Starting with input:
(VOLTAGE INPUT GROUND) = ⇑ <>
   Premise.
(VOLTAGE C3 INPUT) = ⇓ <[Q1 VB] (Q1 ON)>
   KVL-heuristic [Q1 VB]
```

Since (VOLTAGE <n1> <n2>) represents the voltage from <n1> to <n2>, this is equivalent to (VOLTAGE C3 INPUT) = ⇑

```
(CURRENT E Q1) = ⇓ <[Q1 VB] (Q1 ON)>
   V ⇒ IE for Q1
```

The convention is that currents flow into devices, away from nodes. Thus, the current flowing out of the emitter of Q1 is increasing.

```
(VOLTAGE C3 GROUND) = ⇑ <[C3 Q1] [Q1 VB] (Q1 ON)>
   KCL-heuristic [C3 Q1]
(CURRENT C Q2) = ⇓ <[C3 Q1] [Q1 VB] (Q2 ON) (Q1 ON)>
   V ⇒ IC for Q2
(CURRENT #2 RL) = ⇑ <[C3 Q1] [Q1 VB] (Q2 ON) (Q1 ON)>
   KCL for node OUTPUT
```

The current through RL is mentioned twice because the resistor model (model 1), uses the current through the #1 terminal in applying Ohm's Law. The #1 terminal, is the upper terminal of RL, and so the current must flow through terminal #2 to reach terminal #1

```
(CURRENT #1 RL) = ⇓ <[C3 Q1] [Q1 VB] (Q2 ON) (Q1 ON)>
   KCL for device RL
(VOLTAGE OUTPUT +VCC) = ⇑ <[C3 Q1] [Q1 VB] (Q2 ON) (Q1 ON)>
   ⇔ V I for RL
Also given that:
(VOLTAGE +VCC GROUND) = 0 <>
   SUPPLY1
The combination of events (VOLTAGE GROUND +VCC) (VOLTAGE OUTPUT +VCC) cause:
(VOLTAGE OUTPUT GROUND) = ⇑ <[C3 Q1] [Q1 VB] (Q2 ON) (Q1 ON)>
   KVL applied to nodes OUTPUT +VCC GROUND
```

The input voltage is applied to the emitter junction of Q1 causing an increase in its emitter current thereby pulling up the voltage on the emitter. This reduces the base-emitter voltage of Q2 causing its collector current to decrease. Since the current flows through RL, the output voltage drops. This causal argument makes the assumption that the increased input voltage appears across Q1, and that the emitter of Q2 and the collector of Q3 behave as a positive resistance. The first assumption is a result of applying the KVL-heuristic and the second assumption is the result of applying the KCL-heuristic. Because the KCL-heuristic is not applied to the output node, Q2's collector current can be used to deduce the output voltage without making an assumption. Since no signal is ever detected around Q3, all the circuit quantities around Q3, RB1 and RB2 are presumed to be zero. A transistor with no incremental collector current must be fulfilling some quiescent role. In this example Q3 is functioning as a current source.

The complexity of a causal argument depends on the device models used in the analysis. The simpler ideal diode model is sufficient to analyze most circuits. For example, the ideal diode model can explain the DTL-inverter's output behavior. Beta is not easily controlled in transistor fabrication, and so few circuits depend critically on it. For these circuits beta can usually be presumed to be infinite with the base current always zero. Some circuits, notably TTL gates, depend on a fourth region of operation of the transistor. The inclusion of this state unnecessarily complicates the analysis of other circuits, most of which do not depend on it. Since the incorrect choice of oversimplified models usually results in a failure to explain the behavior or an unretractable contradiction, the analysis can always start with simpler models and introduce the more sophisticated models if problems are encountered.

4.9 Quiescent Analysis

People will often include quiescent information in their explanations of incremental behavior. Although this quiescent information is important for many types of analysis, it is not relevant to pure IQ analysis. To include quiescent analysis in this research would violate my methodological position of constructing the simplest mechanism that is adequate for the task of causal analysis. Only after the power and limitations of the simple mechanism have been understood does it make any sense to extend it. To extend it before this analysis is complete makes the determination of which part of the extended mechanism is responsible for success or failure on a particular task nearly impossible.

It is somewhat surprising that the IQ connection heuristics and device models require no reference to quiescent values. For example, if there existed a device whose circuit quantities Q, X and Y were related by the equation $Q = XY$, IQ analysis could not succeed without some quiescent analysis. Differentiating the equation gives:

$$dQ = XdY + YdX$$

Thus to determine the IQ value of Q (i.e. dQ) from the IQ values of X (i.e. dX) and Y (i.e. dY) requires some information about the quiescent values of X and Y. In the case of Ohm's Law $V = IR$, R is positive and $dR = 0$ thereby allowing the IQ resistor model to apply without any quiescent analysis.

The computational machinery that QUAL utilizes for IQ analysis is easily adapted for quiescent analysis. Only a different set of devices models have to be used. The algebra still has four values: "+" value is nonnegative, "0" value is zero, "-" value is nonpositive, and "?" value is unknown. The arithmetic of this algebra is given by table 1 ($\Uparrow$ = +, $\Downarrow$ = -). The quiescent model for a resistor is the same as the IQ model (model 1). The quiescent model for an NPN transistor is:

```
If Q is on, vbe = +, vce = +, ib = +, ic = +, ie = -
If Q is off, ib = 0, ic = 0, ie = 0
If Q is sat, ib = +, ic = +, ie = -, vbe = +, vce = +
```

Model 10

KCL apply as usual, but the connection heuristics are unnecessary. These models are determine all the currents in the emitter-coupled pair:

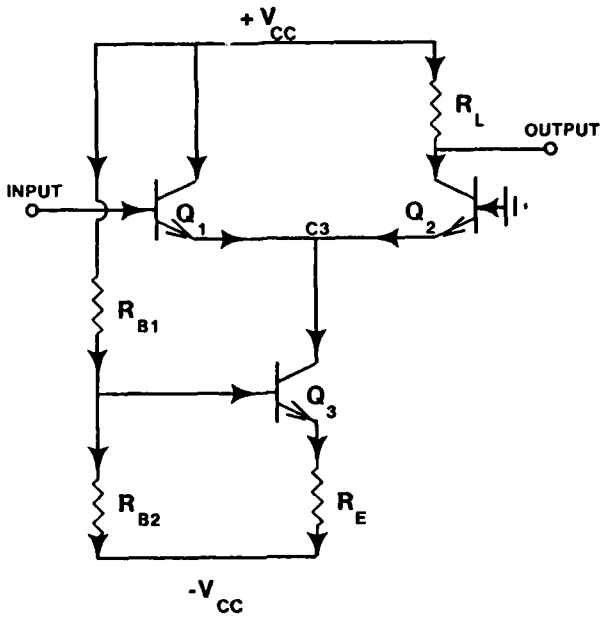


Figure 11 : Emitter-Coupled Pair

Most of the voltages are also determined.

The quiescent analysis is, in general, incomplete. Unlike basic IQ analysis (without the connection heuristics), this quiescent analysis usually is missing only a few currents. Heuristics could be identified to determine the few remaining unknown currents, but this has not been done.

The origin of the confusion about signs is that engineers often pick conventions such that as many circuit quantities as possible are normally positive. For example, an engineer will often define i_C and i_B to be the current flowing into an NPN transistor, and i_E to be the current flowing out of the transistor. With this convention all the quiescent currents are positive. These conventions are then utilized when he gives an IQ explanation. This explanation thus gives all IQ values explicitly and all the quiescent values implicitly. In the following causal argument the sign conventions have been picked (by QUAL) to make all the quiescent quantities positive. CURRENT-INTO and CURRENT-OUT-OF indicate whether the current is flowing into or out of the device.

```
Starting with input:
```

```
(VOLTAGE INPUT GROUND) = ⇑
   Premise.
(VOLTAGE INPUT C3) = ⇑
   KVL-heuristic [Q1 VB]
(CURRENT-OUT-OF E Q1) = ⇑
   V ⇒ IE for Q1
(VOLTAGE C3 GROUND) = ⇑
   KCL-heuristic [C3 Q1]
(CURRENT-INTO C Q2) = ⇓
   V ⇒ IC for Q2
(CURRENT-OUT-OF #2 RL) = ⇓
   KCL for node OUTPUT
(CURRENT-INTO #1 RL) = ⇓
   KCL for device RL
(VOLTAGE +VCC OUTPUT) = ⇓
   ⇔ V I for RL
Also assuming that:
(VOLTAGE +VCC GROUND) = 0
   SUPPLY1
The combination of events (VOLTAGE GROUND +VCC) (VOLTAGE OUTPUT +VCC) cause:
(VOLTAGE OUTPUT GROUND) = ⇑
   KVL applied to nodes OUTPUT +VCC GROUND
```

This section is intended to clarify the relationship between incremental and quiescent analysis, and not to introduce new mechanisms. *No other analysis described in this report utilizes any quiescent analysis*, although purely for expository purposes some of the explanations presented in the following chapters will incorporate the above quiescent sign conventions.

4.10 Recognition and Rationalization

The propagator can now generate a possible explanation for how the DTL-inverter works. This explanation is a rationalization, carrying no guarantee that the inverter functions. The DTL-inverter has 12 possible states, and the analysis reveals that if the circuit is an inverter, inversion must take place in the one state where all devices are on. This is a kind of recognition, answering the question "Could x perform function y?" [de Kleer 77]. Moreover, it gives a causal explanation

of how that function could be achieved.

Since circuits can have state, the response of a circuit to a signal can be a transition from one state to another. Individual devices change state when the signals applied to them change, and thus incremental analysis can determine possible state transitions and their causes. Although the stability of these possible states cannot be verified without doing a quiescent analysis, incremental analysis can determine all the possible state transitions the circuit might follow in response to an input signal.

An example of a transition rule for the npn transistor is: if the v_{BE} is increasing and the transistor is off, it may eventually turn on. Rules of this kind fit neatly into the device models:

```
If Q is on, ↕vbe ⇒ ↕ic, ↕vbe -⇒ ↕ie, ↕vbe ⇒ ↕ib
            ⇑↕vbe → sat, ⇓↕vbe → off

If Q is off, ↕ib = 0, ↕ic = 0, ↕ie = 0
             ⇑↕vbe → on

If Q is sat, ↕ic = 0
             ⇓↕vbe → on
```

Model 11

The expression ⇓↕vbe → off indicates that a possible transition to the off state may occur if the signal is falling.

The model for a diode is much simpler:

```
If D is on, ↕v ⇒ ↕i
            ⇓↕v → off

If D is off, ↕i = 0
             ⇑↕v → on
```

Model 12

When these transition models are used in the incremental analysis of the DTL-inverter, four possible state changes are found:

```
TRANSITION-RULE-4
<(D1 . ON)> → <(D1 . OFF)>
Cause:(VOLTAGE INPUT GROUND) = ⇑
```

If the input diode is on, a rising input voltage may eventually cause it to turn off.

```
TRANSITION-RULE-3
<(D2 . OFF) (D1 . ON)> → <(D2 . ON) (D1 . ON)>
Cause:(VOLTAGE N1 GROUND) = ⇑
```

If the input diode is on, its anode must be rising with the input signal. Thus, if the drop diode is off, it may eventually turn on.

```
TRANSITION-RULE-2
<(Q1 . OFF) (D2 . ON) (D1 . ON)> → <(Q1 . ON) (D2 . ON) (D1 . ON)>
Cause:(VOLTAGE BASE GROUND) = ⇑
```

If both diodes are on, the rising input is communicated to the base of the transistor and if it is off it may eventually turn on.

```
TRANSITION-RULE-1
<(Q1 . ON) (D2 . ON) (D1 . ON)> → <(Q1 . SAT) (D2 . ON) (D1 . ON)>
Cause:(VOLTAGE BASE GROUND) = ⇑
```

If both diodes are on, the rising input is communicated to the base of the transistor and if it is on, it may eventually saturate.

Applying these three transition rules to the 12 possible states results in 11 possible transitions between states. The circuit's states are described by (D1's state, D2's state, Q1's state):

```
TRANSITION-11:(ON OFF SAT) → (ON ON SAT)[TRANSITION-RULE-3]
TRANSITION-10:(ON OFF SAT) → (OFF OFF SAT)[TRANSITION-RULE-4]
TRANSITION-9:(ON OFF OFF) → (ON ON OFF)[TRANSITION-RULE-3]
TRANSITION-8:(ON OFF OFF) → (OFF OFF OFF)[TRANSITION-RULE-4]
TRANSITION-7:(ON OFF ON) → (ON ON ON)[TRANSITION-RULE-3]
TRANSITION-6:(ON OFF ON) → (OFF OFF ON)[TRANSITION-RULE-4]
TRANSITION-5:(ON ON SAT) → (OFF ON SAT)[TRANSITION-RULE-4]
TRANSITION-4:(ON ON OFF) → (ON ON ON)[TRANSITION-RULE-1]
TRANSITION-3:(ON ON OFF) → (OFF ON OFF)[TRANSITION-RULE-4]
TRANSITION-2:(ON ON ON) → (ON ON SAT)[TRANSITION-RULE-2]
```

```
TRANSITION-1:(ON ON ON) → (OFF ON ON)[TRANSITION-RULE-4]
```

These state transitions correspond to the following state diagram:

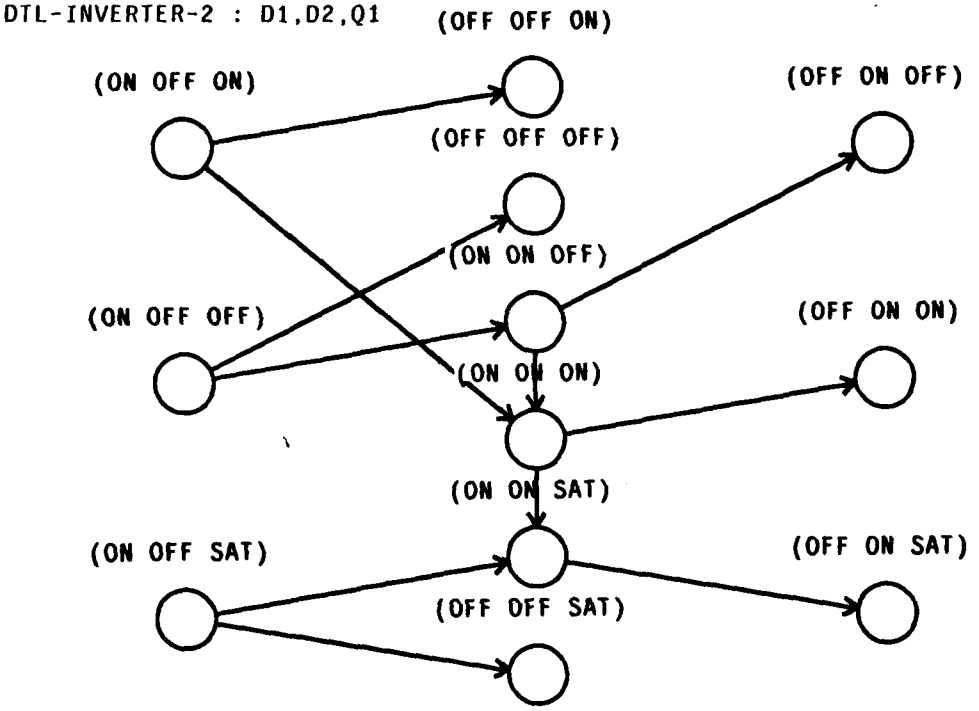


Figure 12 : State Diagram for DTL-Inverter

Any state in which D1 is off has no outgoing transitions, because no signal can be communicated to the rest of the circuit when the input diode is off. The analysis cannot determine whether D1 turns off first or whether D2 turns on first. This is reflected in `TRANSITION-RULE-4` and `TRANSITION-RULE-3`. A quiescent analysis could determine that `(OFF OFF ?)` was an impossible state. If `vcc` is more than two diode drops above ground, current must be flowing through R1 and one of D1 or D2 must be on. Further quiescent analysis could eliminate more of these states, but most of them can be eliminated by applying some simple heuristics to the state diagram.

In order to exhibit useful behavior, a circuit must respond to input signals. This simple *non-autism* rule substantially reduces the state diagram. For example, the state `(OFF OFF OFF)` can be eliminated because it can only be preceded by state `(ON OFF OFF)`, and the output is zero

in both states. The same argument applies to state (OFF OFF SAT). State (OFF OFF ON) is eliminated since it and the preceding state always have a rising signal. The new state diagram is:

DTL-INVERTER-2 : D1,D2,Q1

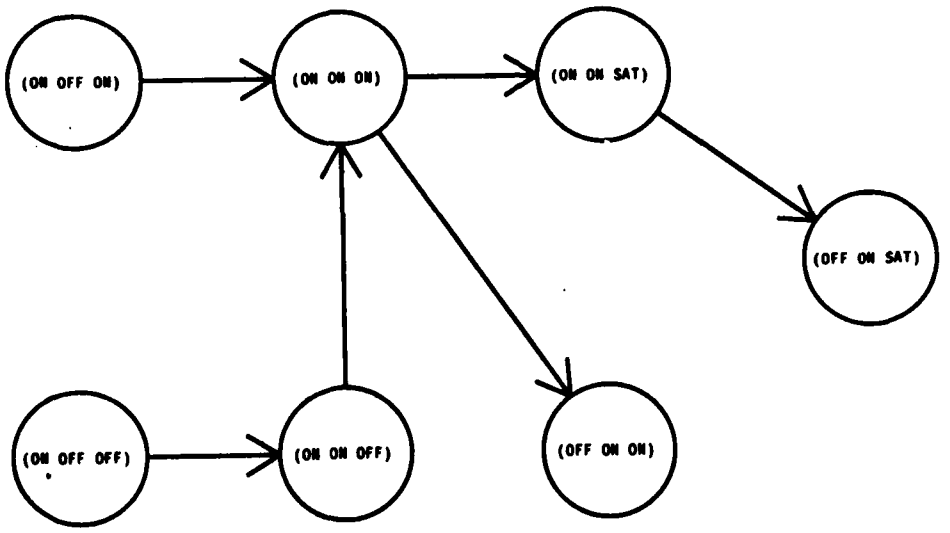


Figure 13 : State Diagram for DTL-Inverter after Simplification

The (ON OFF ON) state is impossible and could be ruled out by a simple quiescent analysis. The (ON OFF OFF) and (OFF ON ON) states can only be ruled out by knowing the DTL-inverter's teleology.

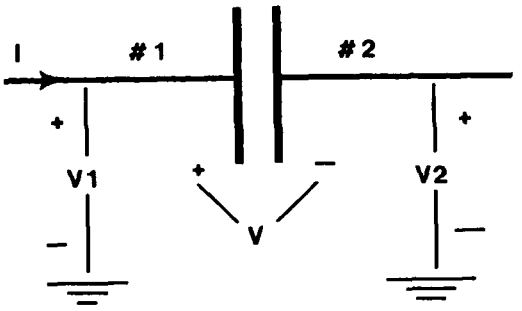
Rules of this kind are insufficient to deal with all behaviors. Fortunately, this is not the goal of this endeavor. Determining a circuit's function solely by analysis is, in general impossible, and rarely interesting. Instead, the unsimplified state diagram can be used to determine whether the circuit could perform a specified function. The DTL circuit is supposed to be an inverter and applying this restriction that the circuit inverts to the original state-diagram (figure 12) also results in the simplified state-diagram (figure 13). The circuit could be a DTL-inverter, and if it is, the analysis has provided a causal explanation for how the circuit achieves that function.

4.11 Capacitance and Elapsed Time

Capacitors differ from the network elements considered so far in that their output behavior depends on their input history as well as their current inputs. Analyzing the behavior of capacitors requires the notion of elapsed time. To some extent this elapsed time can be treated in much the same way as the imposed causal time. The two different kinds of time force causal analysis to make a distinction between instantaneous and long term behavior. This issue is finessed with the transistor, diode and resistor models since their instantaneous and long term behaviors are identical. This is not the case with capacitors and inductors. (Instantaneous and long term behavior correspond approximately to transient and steady state behavior.)

Circuits are designed to function within a particular time scale. Variations in signal of longer or shorter duration than this scale need not be considered. Within this midband most capacitors can either be considered as open or shorted. Bypass and coupling capacitors are shorted in the midband, and parasitic capacitors are open. The type of each capacitor can be determined by doing a causal analysis with each of the capacitor models. A coupling capacitor has to be modeled at least as a resistor or the circuit can have no output. A bypass capacitor has no effect on circuit output, but eliminates any causal changes in the bias network when it is shorted. The parasitic capacitors' only influence is to degrade circuit performance by adding undesired feedback and loading. By running the causal analysis employing the different models the correct usage of the capacitor can usually be determined.

The instantaneous behavior of a capacitor is more complex. A capacitor is quantitatively modeled by a differential equation which relates the current through the capacitor to the derivative of the voltage across it. This behavior can be captured in an IQ model by quantizing the quiescent value into qualitative states. The following is an extremely simple-minded model for a capacitor. The KVL-heuristic is disabled for this model.




```
If C is strt, ↕v1 C⇒ ↕v2, ⇑↕v1 → chg+, ⇓↕v1 → chg-
              ↕v2 C⇒ ↕v1, ⇑↕v2 → chg-, ⇓↕v2 → chg+
              ↕v = 0, ⇑↕i → chg+, ⇓↕i → chg-

If C is chg+, ↕i = ⇓, → strt

If C is chg-, ↕i = ⇑, → strt
```

Model 13

The capacitor is modeled by three states. The state strt represents the situation when the signal is initially applied. The direction of the signal is remembered by a state transition to chg+ or chg-. These states correspond to the situation when the capacitor is charging. (Charging and discharging are indistinguishable in this model.) The capacitor may eventually stop charging and go back into its original state. This transition is not caused by any event, but depends only on the passage of time. The idea behind the model is that the strt state senses the applied impulse which is remembered after the impulse ends. After the impulse ends, the capacitor acts as a current source which represents the current the rest of the circuit is trying to force through it. This model is very simple-minded and fails to capture much of a capacitor's behavior, nevertheless it is sufficient for many circuits.

Consider the following capacitive coupled amplifier stage:

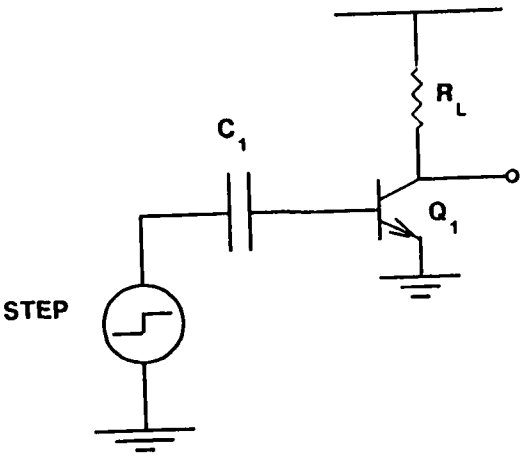


Figure 14 : Capacitor Coupling

Suppose a simple step is applied:

```
If S is step, ↕v = ⇑, → short
If S is short, ↕v = 0
```

The state step models the behavior when the source is changing value, while the state short models the behavior after the source has stopped changing. Behavior before the change is not included in this model. Using the step and capacitor models the instantaneous behavior can be explored by the same techniques used in the previous section. The new causal agent, the passage of time, must be applied wherever possible. In the following analysis no distinction is made between elapsed time and imposed time.

The transition rules are:

```
<(STEP STEP)> → <(STEP SHORT)>
Cause:PASSAGE-OF-TIME
```

The voltage step quickly rises to its final value.

```
<(Q1 OFF) (STEP STEP) (C1 STRT)> → <(Q1 ON) (STEP STEP) (C1 STRT)>
Cause:(VOLTAGE B GROUND) = ⇑
```

The rising voltage step may turn Q1 on.

```
<(Q1 ON) (STEP STEP) (C1 STRT)> → <(Q1 SAT) (STEP STEP) (C1 STRT)>
Cause:(VOLTAGE B GROUND) = ⇑
```

The rising step can force Q1 into saturation.

<(STEP STEP) (C1 STRT)> → <(C1 CHG+) (STEP STEP)>

Cause:(VOLTAGE INPUT GROUND) = ⇑

C1 senses the rising input signal and starts to charge.

<(C1 CHG+)> → <(C1 STRT)>

Cause:PASSAGE-OF-TIME

The capacitor may stop charging and return to its quiescent state.

<(C1 CHG+) (Q1 SAT)> → <(C1 CHG+) (Q1 ON)>

Cause:(VOLTAGE B GROUND) = ⇓

As C1 charges to its final value, it may no longer be able to supply enough current to keep Q1 in saturation.

<(C1 CHG+) (Q1 ON)> → <(C1 CHG+) (Q1 OFF)>

Cause:(VOLTAGE B GROUND) = ⇓

As C1 charges further, it may no longer be able to supply enough current to keep Q1 in its active region.

Applying these transition rules to the possible states results in the state diagram of figure 15. One plausible path through the diagram is indicated with arrows. Assuming that Q1 is normally off, the voltage step applied to state (STRT OFF STEP) first moves Q1 into its active region and then moves it into its saturated region (STRT SAT STEP). C1 senses the charging current (⇑↕i → chg+) and the step rises to its final value (CHG+ SAT SHORT). As C1 charges the voltage at the base drops eventually turning Q1 off (STRT OFF SHORT).

Although the capacitor model coupled with the state transition analysis is able to give a plausible explanation for the circuit's behavior, it did illustrate some serious problems. These are subjects for further research.

Most circuits have a single state which they eventually return to after being perturbed. For example, if the stable state is (? OFF ?), the states (STRT SAT OPEN) and (STRT ON OPEN) can be ruled out.

QUAL does not allow two state changes to happen simultaneously. But (STRT SAT STEP) → (CHG+ SAT OPEN) is quite plausible.

There is no notion of time-constant. For example, the time to charge C1 and the rise time of the voltage step cannot be compared. This causes serious problems when the circuit contains

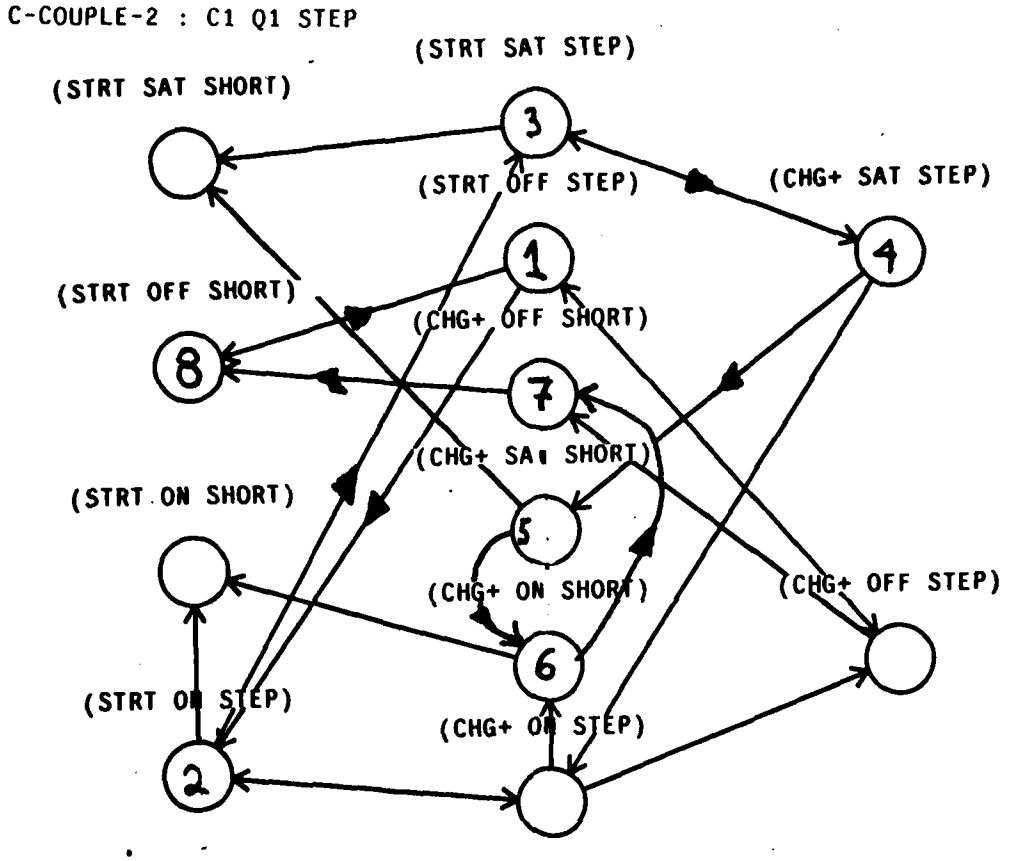


Figure 15 : State Diagram for Capacitive Coupling

more than one capacitor or inductor.

When a monotonic signal is charging a capacitor it cannot return to its quiescent state. For example, the oscillation (STRT ON STEP) ↔ (CHG+ ON STEP) suggested in the diagram is electrically impossible.

The device model for a transistor had to be modified to make the analysis possible. When a transistor is off it has zero base current making it impossible to charge the capacitor. For this example, the transistor model was modified to always have finite input impedance.

The model is electrically inaccurate since a capacitor becomes open after a sufficient amount of time has elapsed.

Some of the transitions are optional and some are mandatory. Whether or not the transition to Q1 on occurs depends on the height of the step. The transition of the transistor back to its

original state is mandatory since it must eventually stop charging.

4.12 The Relationship Between Causality and Constraint

There is an interleaved hierarchy of causal and constraint-like descriptions for the same physical phenomena. In the Schmitt trigger description the engineer uses a causal description. In order to determine the precise values of the electrical quantities he will employ a constraint representation consisting of algebraic equations. This lumped-parameter representation is modeled on more basic causal phenomena.

In contrast to causal arguments, a quantitative description of a system's behavior is in terms of a set of quasistatic constraints describing the dynamics of the system. Indeed, the lumped-parameter circuit model of the physical system described by a circuit diagram is only valid under the assumption that the system is always at equilibrium. But a circuit is only useful because its equilibrium changes under the influence of imposed signals. The "force" that moves a circuit from one interesting equilibrium to another, when driven by a signal, is that the incremental signal slightly displaces the equilibrium from the circuit's state. The process of equilibrating is adequately described by the differential equations of the dynamics of the circuit. The manner in which the signal moves the equilibrium around is better described by the qualitative, causal arguments.

Although the equilibrating process can be quantitatively described, it is difficult to quantitatively describe the manner in which the signal moves the equilibrium. The lumped-parameter circuit model is an idealization and simplification of the behavior of the electromagnetic fields in and around the circuit components. Since changes in these fields propagate at finite speeds, this process takes a certain amount of time. The differential equations of the lumped-parameter circuit model cannot account for what happens during this period of disequilibrium. Within this period the changing fields of the input signals propagate until global equilibrium is reached. This propagation can be viewed as a kind of causal flow: the input field changes and propagates to other materials causing further fields to change. These changes can be partially ordered in a time sequence in which each change is caused by changes earlier in the sequence and earlier in time.

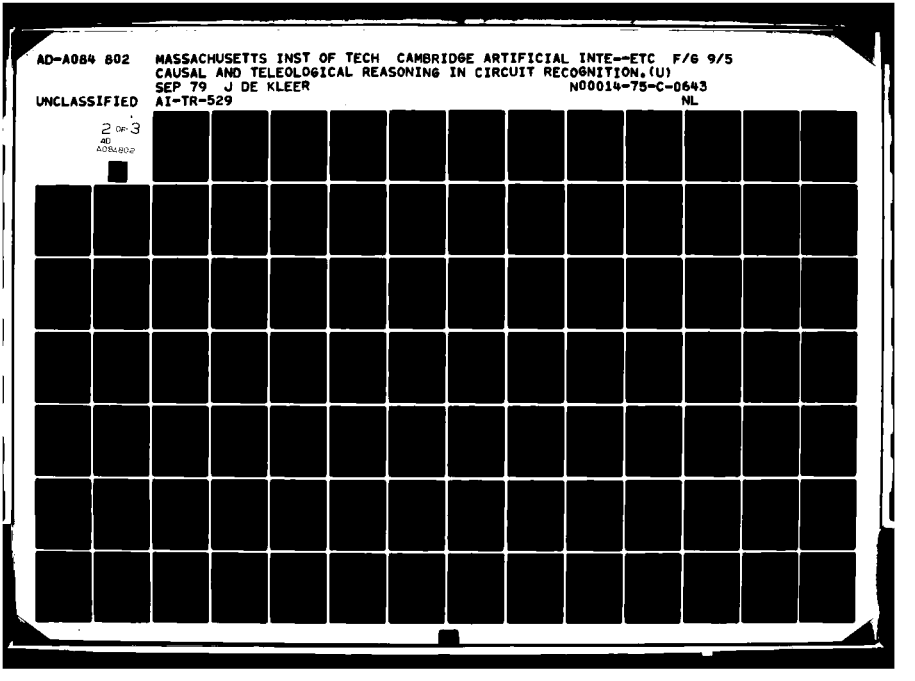
The quantitative calculation of the causal flow that happens during the period of disequilibrium is intractable. Although the electromagnetic laws that govern the physics are known, there is no practical way to quantitatively describe this causal process. This means that the electrical engineer

can never completely analyze a circuit. Fortunately the engineer is only interested in analyzing the circuit to a certain amount of precision and the lumped-circuit model provides a technique for this. If the disequilibrium in a particular area of the circuit is important to the overall behavior, the engineer introduces *parasitic* capacitors and inductors to describe this disequilibrium. This technique captures the quasistatic effects of the disequilibrium but not the causal effects.

When an electrical engineer reasons about a circuit he tries to reintroduce the causality that the lumped-circuit model throws away. He does this by using locally causal models and imposing a time flow on the changes in circuit quantities. Only by throwing away most of the detail of the models and the causality is he able to make the causal analysis tractable. The engineers' qualitative theory of circuit causality explains the period of equilibration by introducing finite time flow and permitting the circuit to be in disequilibrium. The actual lumped-circuit model of the circuit he uses allows him to include only those effects that are important, and his causal argument describes the effect of each component in the disequilibrium period.

AD-A084 802 MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTE--ETC F/G 9/5
CAUSAL AND TELEOLOGICAL REASONING IN CIRCUIT RECOGNITION.(U)
SEP 79 J DE KLEER N00014-75-C-0643
UNCLASSIFIED AI-TR-529 NL

2 OF 3
AD
A084802



Chapter 5

INTERPRETATIONS

5.1 Points of View on Circuit Behavior

In the process of causal analysis the propagator makes assumptions and, as a consequence, may discover different values for the same circuit quantity. The problem of determining which of these values is the correct one reduces to choosing among the assumptions underlying the different values. Since the same assumption can underlie many circuit quantities, a selection of a value for one circuit quantity may automatically force the selection of other circuit quantities. In order to avoid erroneous choices, all possible consequences of assumptions must be included in the selections. The disambiguation of circuit quantities thus becomes a global computation on assumptions, not values.

The computational origin of the ambiguities is the fact that the causal analysis has "simulated" the circuit in all possible ways simultaneously. Although, all the events and causal interactions of all the possible causal arguments have been discovered, the analysis has not separated the events into the different causal arguments. If every assumption in the causal simulation corresponded to a fork in a partial causal argument, the complete arguments could be identified by examining the terminal events. Unfortunately the dependency relationships among the events form a general graph structure rendering the trivial strategy useless. Instead, the identification of tentative causal arguments involves finding collections of assumptions which select the events of the causal arguments. Such a collection of assumptions is called an *interpretation.* This chapter is concerned with establishing what criteria a collection of assumptions must meet to be considered an interpretation. For example, one such criterion is that the interpretation may not select contradictory values for an event in the causal argument (an interpretation selects a value if the assumptions of the value are a subset of assumptions of the interpretation). Chapter 7 addresses the problem of determining which of the clearly identified causal arguments is the correct one.

Two important definitions of interpretation will be presented. One definition of interpretation is useful for fault localization and the other is appropriate for recognition. The problem of

fault localization motivates the idea of utilizing reverse causal reasoning to identify causes for undesirable behavior. The second definition leads to a technique for expressing the causal flow in the circuit as an acyclic graph. This *causal* graph serves as the basis for the recognition procedures discussed in the next three chapters.

5.2 Measurement Interpretations

The analysis process usually discovers multiple values for the circuit quantities. If two of these values differ and have compatible environments, a contradiction is recorded. Note that a value has the three parts: IQ expression (e.g. ⇓), environment (e.g. `<[B1 IN] (Q1 ON)>`) and derivation (e.g. `V` ⇒ `IC for Q1`). In the case where one of the environments is a subset of the other, one or both of these values will immediately stop propagating. Although contradictions rule out most of the multiple values, many cells still contain multiple, possibly differing values at the conclusion of the analysis. If all of the values in a particular cell are the same, then no further analysis is necessary since the value holds independently of any environment. However, if the values differ, the correct environment needs to be disambiguated in order to determine the correct value (unless the circuit has multiple stable states).

The IQ expressions and environments of the circuit's output values form insufficient evidence upon which to base the disambiguation. Even if all of the outputs had the same IQ expression, they may have been generated by different causal arguments. Furthermore a causal argument for the output does not necessarily reference all of the circuit's components. For example, bias networks and loads enable the signal path to exist, but do not have any IQ contribution to the signal. Feedback paths are also not mentioned in the causal argument for the output. The assumptions underlying the causal arguments characterize the fact that all of these components are working appropriately. Therefore in order to arrive at the different causal arguments every circuit value must be considered, not just those directly on the signal path. This criterion requires that an interpretation must be a maximal collection of assumptions: no assumption can be added to an interpretation without violating one of the other criteria. This maximality condition ensures that as many circuit quantities are selected by the interpretation as possible. In a few cases the local causal flow in a component is indeterminate, and to handle these cases the interpretation may contain incompatibilities (e.g. an interpretation might contain both [RF V1] and [RF V2]).

At the conclusion of the analysis those cells which have not received values are presumed to

contain zero. The rationale is that an effect must have a cause, and all possible causes have been explored. The connection heuristics make the implicit assumption that all unknown quantities are zero, so there is no necessity for propagating these values. An interpretation may select no value for a cell and thus take advantage of the fact that a cell with no values is zero. If a cell has no values under a certain interpretation, no cause has been found for it, and therefore it is presumed to be zero.

The process of causal analysis explores all possible interpretations of a circuit's behavior. Although it is good at determining causal arguments within a particular interpretation, it is bad at identifying which interpretation is the correct one. Some assumptions can be verified by causal reasoning and other assumptions critically depend on parameter values, but the verification of most assumptions requires fundamentally different analysis techniques. The latter are based on more complicated reasoning about constraints and purposes. One way to avoid applying these techniques is to build the circuit and take measurements.

Two of the four arguments causal analysis finds for the output behavior of the feedback amplifier are:

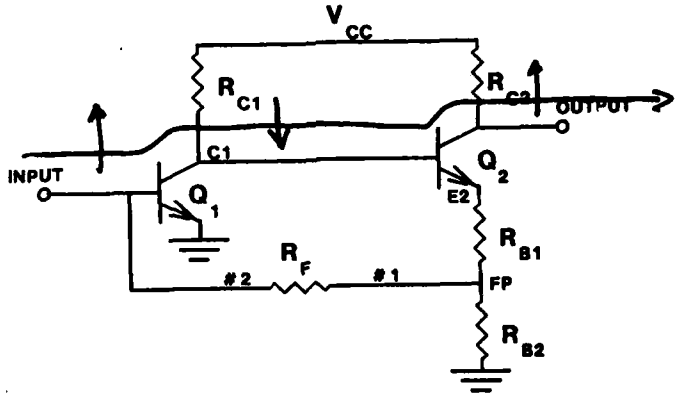


Figure 1 : Feedback Amplifier

```
Starting with input:
(CURRENT-INTO TERMINAL IN) = ⇑ <>
   Premise.
(VOLTAGE B1 GROUND) = ⇑ <[B1 IN]>
   KCL-heuristic [B1 IN]
```

```
(CURRENT-INTO C Q1) = ⇑ <[B1 IN] (Q1 ON)>
   V ⇒ IC for Q1
(VOLTAGE C1 GROUND) = ⇓ <[C1 Q1] [B1 IN] (Q1 ON)>
   KCL-heuristic [C1 Q1]
(VOLTAGE C1 E2) = ⇓ <[Q2 VB] [C1 Q1] [B1 IN] (Q2 ON) (Q1 ON)>
   KVL-heuristic [Q2 VB]
(CURRENT-INTO C Q2) = ⇓ <[Q2 VB] [C1 Q1] [B1 IN] (Q2 ON) (Q1 ON)>
   V ⇒ IC for Q2
(CURRENT-OUT-OF #2 RC2) = ⇓ <[Q2 VB] [C1 Q1] [B1 IN] (Q2 ON) (Q1 ON)>
   KCL for node OUTPUT
(CURRENT-INTO #1 RC2) = ⇓ <[Q2 VB] [C1 Q1] [B1 IN] (Q2 ON) (Q1 ON)>
   KCL for device RC2
(VOLTAGE VCC OUTPUT) = ⇓ <[Q2 VB] [C1 Q1] [B1 IN] (Q2 ON) (Q1 ON)>
   ⇔ V I for RC2
Also given that:
(VOLTAGE VCC GROUND) = 0 <>
   SUPPLY
The combination of events (VOLTAGE GROUND VCC) (VOLTAGE OUTPUT VCC)
      cause:
(VOLTAGE OUTPUT GROUND) = ⇑
      <[Q2 VB] [-C1 Q1] [+B1 IN] (Q2 ON) (Q1 ON)>
   KVL applied to nodes OUTPUT VCC GROUND
```

The increased input voltage turns Q1 on harder, pulling down its collector. This falling voltage is applied to the base of Q2, causing it to begin to turn off. Since Q2's collector current is dropping, the voltage across the load RC2 must also drop, causing the output to rise.

The second causal argument is:

```
Starting with input:
(CURRENT-INTO TERMINAL IN) = ⇑ <>
   Premise.
(VOLTAGE B1 GROUND) = ⇑ <[B1 IN]>
   KCL-heuristic [B1 IN]
(VOLTAGE FP B1) = ⇓ <[RF V2] [B1 IN]>
   KVL-heuristic [RF V2]
```

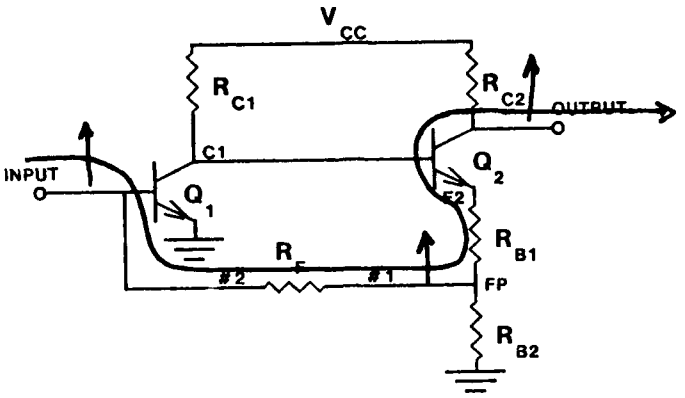


Figure 2 : Feedback Amplifier

```
(CURRENT #1 RF) = ⇓ <[RF V2] [B1 IN]>
   V ⇒ I for RF

(VOLTAGE FP GROUND) = ⇑ <[FP RF] [RF V2] [B1 IN]>
   KCL-heuristic [FP RF]

(VOLTAGE E2 FP) = ⇓ <[RB1 V2] [FP RF] [RF V2] [B1 IN]>
   KVL-heuristic [RB1 V2]

(CURRENT-INTO #1 RB1) = ⇓ <[RB1 V2] [FP RF] [RF V2] [B1 IN]>
   V ⇒ I for RB1

(VOLTAGE E2 GROUND) = ⇑ <[E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN]>
   KCL-heuristic [E2 RB1]

(VOLTAGE C1 E2) = ⇓
      <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   KVL-heuristic [Q2 VE]

(CURRENT-INTO C Q2) = ⇓
      <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   V ⇒ IC for Q2

(CURRENT-OUT-OF #2 RC2) = ⇓
      <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   KCL for node OUTPUT

(CURRENT-INTO #1 RC2) = ⇓
      <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   KCL for device RC2

(VOLTAGE VCC OUTPUT) = ⇓
      <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   ⇔ V I for RC2
```

```
Also assuming that:
(VOLTAGE VCC GROUND) = 0 <>
   SUPPLY
The combination of events (VOLTAGE GROUND VCC) (VOLTAGE OUTPUT VCC)
      cause:
(VOLTAGE OUTPUT GROUND) = ⇑
      <[Q2 VE] [E2 RB1] [RB1 V2] [FP RF] [RF V2] [B1 IN] (Q2 ON)>
   KVL applied to nodes OUTPUT VCC GROUND
```

The increased input voltage is coupled through RF and RB1 to the emitter of Q2. The rising voltage at the emitter causes Q2 to begin to turn off, consequently lowering its collector current. The voltage across the load RC2 must also drop.

The circuit's behavior has four interpretations:

```
<[B1 IN] [RB1 V1] [RF V2] [FP RF] [RB1 V2] [E2 Q2] [Q2 VB] [C1 Q1]>
<[B1 IN] [Q2 VE] [RF V2] [FP RF] [RB1 V2] [E2 RB1] [Q2 VB] [C1 Q1]>
<[B1 IN] [RF V2] [FP RF] [RB1 V2] [E2 RB1] [Q2 VE] [C1 Q2]>
<[B1 IN] [RF V1] [E2 Q2] [RF V2] [RB1 V1] [FP RB1] [Q2 VB] [C1 Q1]>
```

The four interpretations originate from the circled ambiguities:

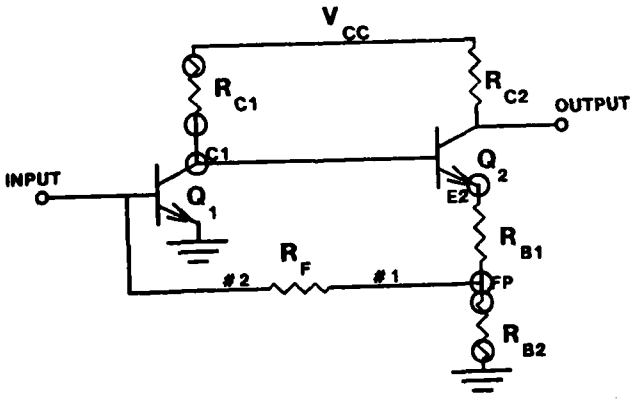


Figure 3 : Feedback Amplifier Ambiguities

An ambiguity is *minimal* if there is no other simpler ambiguity whose resolution would automatically resolve it as well. Two ambiguities are *similar* if they involve identical environments. If only the minimal instance of each ambiguity is retained, and if similar ambiguities are grouped together, only three ambiguities remain:

```
Ambiguity between
<[B1 IN] [E2 RB1] [RB1 V2] [FP RF] [RF V2]> (2)
<[B1 IN] [E2 Q2] [Q2 VB] (Q2 ON) [C1 Q1] (Q1 ON)> (1)
occurs at
(VOLTAGE E2 GROUND)
(VOLTAGE E2 VCC)

Ambiguity between
<[B1 IN] [FP RF] [RF V2]> (3)
<[B1 IN] [FP RB1] [Q2 VB] (Q2 ON) [C1 Q1] (Q1 ON)> (1)
occurs at
(VOLTAGE FP GROUND)
(VOLTAGE FP VCC)
(CURRENT #1 RB2)
(CURRENT #2 RB2)

Ambiguity between
<[B1 IN] [C1 Q1] (Q1 ON)> (3)
<[B1 IN] [C1 Q2] [Q2 VE] (Q2 ON) [E2 RB1] [RB1 V2] [FP RF] [RF V2]> (1)
occurs at
(VOLTAGE C1 GROUND)
(VOLTAGE C1 VCC)
(CURRENT #1 RC1)
(CURRENT #2 RC1)
```

Through careful analysis of the environments, the number of measurements required to resolve the ambiguities can be minimized. Each measurement will contradict one of the two environments of an ambiguity. The contradiction of any particular environment may also automatically resolve other ambiguities. For example, if environment <[B1 IN] [C1 Q1] (Q1 ON)> of the third ambiguity is contradicted, all the other ambiguities are automatically resolved since each other ambiguity has one environment which contains <[B1 IN] [C1 Q1] (Q1 ON)>. This number is indicated after the environment in the summary. Since there is no *a priori* information about which environments will be contradicted, the next ambiguity to resolve is selected on the basis of the average number of ambiguities that would be resolved by the measurement. By this measure the second and third ambiguities have better scores, and QUAL arbitrarily picks the second. Voltage measurements

are usually easier to take, so QUAL asks for one of the two voltages that would resolve the ambiguity:

```
Optimal voltage measurements are:
(VOLTAGE FP VCC)
(VOLTAGE C1 VCC)
Is the value of (VOLTAGE FP VCC) ⇓ or ⇑
```

The voltage at FP is observed to be falling. The final consistent interpretation for the circuit's behavior is:

```
<[B1 IN] [RF V1] [E2 Q2] [RF V2] [RB1 V1] [FP RB1] [Q2 VB] [C1 Q1]>
```

Since all the possible assumptions about RF are included in this interpretation, the causality around RF has not been clarified. This interpretation identifies the correct causal argument for the output presented in figure 1. (Unfortunately, since VCC is incrementally GROUND this simple strategy weights measurements unfairly.)

Since causal analysis did not determine the correct interpretation, the selected causal argument remains a rationalization of the observed behavior. Causal analysis assigns multiple values to circuit quantities only if they can be derived in multiple ways. This only happens if the circuit contains possible feedback paths. Since ambiguities always stem from possible feedback paths, explicit knowledge about feedback should be incorporated into the analysis process.

5.3 Fault Localization

This thesis has described the beginnings of a theory of what it means to understand how a circuit works. One test of such a theory must be whether this understanding can be utilized to analyze circuit faults. One use of fault localization techniques is *troubleshooting*. Troubleshooting involves determining why a particular correctly designed circuit is not functioning as intended, the explanation for the faulty behavior being that the particular instance of that circuit under consideration is at variance in some way with its design. The same techniques are also applicable to *debugging* almost correct designs [Sussman 77]. If the designer has a description of how the circuit should behave and has an implementation of that behavior that is correct except for some small local problem, the intentions of the designer can be used to determine which component is contributing to the unintended behavior.

The success of the fault localization strategies discussed here will depend on having a description of how the circuit should work and on whether the fault is localized to a small area of the circuit.

Every mechanism which can predict behavior can be utilized to predict the new behavior which would result if a fault were introduced. Troubleshooting by synthesis exhaustively hypothesizes all possible faults and eliminates those faults which are not consistent with the observed symptoms. This technique is computationally impractical with conventional circuit analysis programs. Moreover, special techniques have to be developed to cope with the infinite number of faults a single component can have (e.g. a resistor can have an infinite number of incorrect values). Since causal analysis uses a simple algebra, the computation is more tractable and the number of faults a component can have is limited.

There are two different techniques for evaluating hypothetical faults. A faulty model can be used in the usual causal analysis to determine whether the predicted behavior is consistent with the observed symptoms. This technique would determine that a failing DTL-inverter could be explained by D1 being stuck off. Another technique is to remove the input signal and treat the faulty change in a model's parameter as the signal. This technique predicts the change in quiescent behavior. If the predicted change is consistent with the difference between the correct and observed quiescent behavior, the fault explains the symptoms. For example, suppose the beta of the DTL-inverter output transistor is too low. Introducing this fault in the state when all the devices are on, the collector current decrements. This explains the quiescent fault that the inverter is not pulling down hard enough. The latter technique is particularly useful in identifying faults in the quiescent aspects of the circuit behavior, and the former technique is useful for identifying faults in the incremental behavior. Unfortunately, neither technique provides a method for making hypotheses. Troubleshooting by synthesis using these evaluation techniques is inefficient both in terms of computational resources and in the number of measurements required to isolate the faulted component. (These strategies become more useful on abstract descriptions of circuits [Brown 76].)

Since causal analysis usually finds multiple interpretations for the behavior, these two techniques work considerably better when the correct interpretation is known. The interpretation can be used to indicate which states to examine for symptomatic behavior. For example, D1 stuck off explains the circuit's inability to invert only if inversion takes place in the state in which

all the devices are on. The more that is known about the circuit's behavior, the easier it is to troubleshoot it. The expected input-output behavior is necessary to determine that a fault exists at all, and knowledge of the correct interpretation guides the analysis of hypothetical faults. By making random measurements, troubleshooting by synthesis will eventually localize the fault, but it is more profitably used as an hypothesis evaluator for the localization strategies.

The interpretation also provides a causal explanation for how the outputs are caused by the inputs. The devices mentioned in this explanation are prime candidates for possible faults and the fault modes can be determined by examining the argument. The resulting hypotheses can be evaluated to determine which faults in which of these devices are consistent with the symptoms. The difficulty with this is that the interpretation may be changed by the presence of the fault. If a causal assumption is violated, the entire argument may be invalidated because the dominant effect may actually be the quantity which caused the violation. Since the designer never intended that the circuit behave in that way, no appeal can be made to the original intention. Similarly, state diagram heuristics which apply to working circuits cannot be used. The interpretation under which the faulty circuit is behaving must be disambiguated by actual measurements. The procedure presented in the previous section can determine the interpretation by taking appropriate measurements. When the new interpretation is identified, its causal argument can be examined for faults.

If the behavior prediction mechanism is invertible, this property can be utilized for fault localization and for design; the symptomatic or desired input-output behavior is used as an input to the inverted prediction mechanism in order to identify faults in or constraints on the individual components. Numerical techniques are not invertible and therefore inapplicable. Propagation of symbolic constraints can be quite successful in synthesizing a circuit from a desired input-output behavior [de Kleer & Sussman 78], but it is not as applicable to troubleshooting [de Kleer 75]. When desired behavior differs from expected behavior, blame can be assigned to any device involved in the propagation. In a detailed analysis the desired output behavior may depend on every circuit device, therefore the observed symptom provides no information. The strategy only becomes informative after internal measurements have been taken which introduce sufficient redundancy that the constraints do not need to depend upon every device in the circuit. Even after some internal measurements have been taken, the strategy is incapable of suggesting further measurements to take. Some other mechanism must be employed to suggest informative

measurements.

A particular causal argument can be inverted to determine what could have caused the undesirable output. However, far more profit can be made by inverting the causal analysis process itself. The direction of time flow can be reversed in the analysis process in order to determine what could have caused the undesirable behavior. The direction of time flow is primarily provided by the models, and these can be easily inverted. For example, an increased transistor v_{BE} is used to derive an increased i_C but not vice versa. When the direction of time flow is reversed, increased i_C is used to derive an increased v_{BE} but not vice versa. In forward time a deduction "A implies B" signifies "A causes B" while in reverse time it signifies "A can be caused by B." The inverted model for a transistor is:

```
if Q is on, ↕ic ⇒ ↕v , ↕ie -⇒ ↕v, ↕ib -⇒ ↕v
if Q is off, ↕ib = 0, ↕ic = 0, ↕ie = 0
if Q is sat, ↕v = 0, ↕ic = 0
```

The other device models are easily inverted.

KCL and KVL remain unchanged. The connection heuristics require major modification. The KVL-heuristic is easily dealt with. In forward time analysis a device model can be triggered by a voltage-to-reference on one of its input nodes. The reverse time KVL-heuristic deduces a voltage-to-reference whenever the inverted device model determines a voltage on an input. For example, the forward KVL-heuristic triggers the transistor rule on the assumption that the base voltage is dominant, and the reverse KVL-heuristic deduces the voltage on the base from the collector current under the assumption that the base voltage was the dominant input that caused the collector current. The assumption is recorded as [Q v_B] in both cases. The reverse KVL-heuristic deduces two incompatible voltages as a consequence of a collector current: an emitter voltage based on assumption [Q v_E] and a base voltage based on assumption [Q v_B].

In order to understand the reverse KCL-heuristic reconsider the network theory behind the assumption:

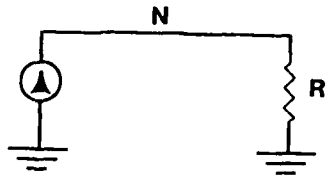


Figure 4 : KCL-heuristic Network Assumption

The forward KCL-heuristic makes the assumption that the unknown currents into a node behave as a positive resistance so that it can predict the voltage at that node. The reverse KCL-heuristic is derived by applying Ohm's law to this positive resistance. In reverse analysis, when a voltage at a node is discovered, it is assumed to be the result of current flowing through the positive resistance. The reverse KCL-heuristic is implemented by a procedure which is triggered whenever a voltage-to-reference is discovered and then assumes that the unknown terminal currents individually receive the current which Ohm's law predicts the entire unknown bundle of terminals should receive. The reverse KCL-heuristic is as complicated as the forward KCL-heuristic because it must carefully analyze the environments of the voltage and the currents.

When time is reversed, the ambiguities in the analysis result from the inability to identify the specific cause for an effect. These ambiguities can be handled by the same methods used in forward time reasoning to handle causes with uncertain effects. Compare the following reverse time causal analysis of the DTL-inverter with the earlier forward time causal argument.

```
Starting with input:
(VOLTAGE OUTPUT GROUND) = ⇑ <>
   Premise.
Also given that:
(VOLTAGE GROUND VCC) = 0  <>
   POSITIVE-SUPPLY
The combination of events (VOLTAGE GROUND VCC) (VOLTAGE OUTPUT GROUND)
      cause:
(VOLTAGE OUTPUT VCC) = ⇑  <>
   KVL applied to nodes GROUND VCC OUTPUT
(CURRENT #1 RL) = ⇓  <>
   ⇔ V I for RL
```

```
(CURRENT #2 RL) = ⇑  <>
   KCL for device RL
(CURRENT C Q1) = ⇓  <>
   KCL for node OUTPUT
(VOLTAGE B GROUND) = ⇓ <(Q1 ON)>
   IC ⇒ V for Q1
(CURRENT #2 D2) = ⇑  <[B D2] (Q1 ON)>
   KCL-heuristic [B D2]
(CURRENT #1 D2) = ⇓  <[B D2] (Q1 ON)>
   KCL for device D2
(VOLTAGE B N) = ⇑  <[B D2] (Q1 ON)>
   I ⇒ V for D2
(VOLTAGE N GROUND) = ⇓  <(D2 ON) [D2 V1] [B D2] (Q1 ON)>
   KVL-heuristic [D2 V1]
(CURRENT #1 D1) = ⇑  <[N D1] (D2 ON) [D2 V1] [B D2] (Q1 ON)>
   KCL-heuristic [N D1]
(VOLTAGE N INPUT) = ⇑  <(D1 ON) [N D1] (D2 ON) [D2 V1] [B D2] (Q1 ON)>
   V ⇒ I for D1
(VOLTAGE.INPUT GROUND) = ⇓
      <[D1 V2] (D1 ON) [N D1] (D2 ON) [D2 V1] [B D2] (Q1 ON)>
   KVL-heuristic [D1 V2]
```

This explanation lists the events in the usual order of discovery that was used for the forward time explanations. Event A is an antecedent of event B if B takes part in a possible causal deduction of A. Note that the above explanation is just the inverse of the forward time explanation.

The forward time flow analysis is an information-losing process; any value which is not an output and does not propagate is lost. Contradictions have no direct effect on the output signal and are also lost. Since the reverse time analysis is given only one piece of information about the forward time flow behavior, it cannot analyze the entire circuit. However, it should be able to find a causal argument to explain what inputs could have caused observed outputs. (It could use forward time flow analysis to check the interpretation it discovers, but QUAL currently does not.)

In the reverse time analysis, the faulty output can be caused by either faulty inputs to the component or the component itself. Just as in troubleshooting by synthesis, there are two different techniques for using the strategy. The undesirable difference between observed and expected quiescent behavior can be treated as the quantity to be explained. Using this technique, a low collector current is explained by a low beta. The other technique explains the undesired response to the applied signal directly, explaining a positive gain in the inverter by a possible base-collector short in the output transistor. Since most faults manifest themselves quiescently, the technique which focuses on the difference between observed and expected quiescent behavior is generally more useful.

Except for assumptions at the external connections the interpretation for a behavior is independent of the direction of time flow. Nevertheless, a fault may force a different unintended interpretation. In order to disambiguate the interpretations, measurements internal to the circuit must be taken. Applying this strategy the fault localization process takes circuit measurements for two purposes. When a possible causal explanation for the symptomatic behavior is known, measurements are necessary to determine which device in this explanation could be faulted. If no interpretation is known, or if measurements invalidate an interpretation, measurements must be taken to determine a new interpretation.

The reverse time localization process is considerably different than troubleshooting by synthesis. It makes only one analysis with the undesired behavior as the input signal, while troubleshooting by synthesis has to do a separate analysis for every possible fault. Although the causal argument they both eventually arrive at to explain the symptomatic behavior is isomorphic (under time-reversal), the reverse time strategy has made a more efficient set of measurements and is able to explain why the measurements were made and why other faults were not considered. The only explanation troubleshooting by synthesis can provide to explain why a device is not faulted is that a fault in the device is not consistent with the observed symptoms. It is also poor at suggesting further measurements.

Both localization strategies are successful, and their success is due in large part to utilizing the knowledge of how the circuit works. The missing piece of the theory is hierarchy. The strategies discussed in this section apply to any level of detail, but they do not explain how to move between levels of detail. An improved localization system would first analyze the fault at the shallowest level of detail. After the fault has been localized to particular modules, it would

consider the implementation of only those models which could contain the fault. (The current implementation does not utilize this hierarchy.)

5.4 Intrinsic and Extrinsic Descriptions of Behavior

The IQ device rules specify the behavior of a component in all possible situations. Causal reasoning employs these specifications of the *intrinsic* behavior of the individual components to determine the behavior of the composite circuit. As part of a circuit a component plays a specific role in behavior of the composite circuit. The *extrinsic* description explains how the intrinsic behavior of the component contributes to the behavior of the circuit. To develop a theory of extrinsic descriptions, the global mechanisms by which circuits achieve their behavior must be examined. Feedback, the single most important such mechanism is discussed in the next chapter. This section lays the foundation for description and recognition by exploring two particular types of extrinsic behavior.

After causal reasoning has discovered the behavior of the composite circuit, the way each component's behavior contributes to this composite behavior provides the basis for the extrinsic descriptions. The extrinsic behavior of a component has two aspects: the local way the behavior was used and the contribution of this behavior to the global pattern of interactions that produce the output. The assumption that the base voltage is a dominant input to the transistor is an example of a local extrinsic description. That a component is part of a feedback network to control amplifier gain is an example of an global extrinsic description.

Global extrinsic descriptions of behavior are necessary for two reasons. They provide the basis for describing at a shallower level of detail how the circuit achieves its purpose. In this way reasoning about behavior can be related to more teleological and hierarchical descriptions of circuit behavior. Secondly, for all but the most simple circuits, causal reasoning discovers multiple interpretations for their behaviors. These interpretations can only be disambiguated by examining how the circuit works as a whole. Thus extrinsic descriptions can be utilized to select the most reasonable alternative from these interpretations. These two topics are discussed in the following chapters.

5.5 Causal Interpretations

A measurement interpretation selects consistent values from each circuit cell. Consequently an interpretation identifies particular causal arguments which describe how inputs affect outputs. Measurement interpretations are defined to be maximal, possibly incompatible, collections of assumptions which select consistent values for each cell. With this definition, there always exists a sequence of measurements which can identify the unique interpretation under which the circuit is behaving. The difficulty with this definition is that it does not necessarily identify a unique causal flow everywhere in the circuit. For example, measurements cannot determine the causality around RF in the feedback amplifier analyzed in the previous chapter.

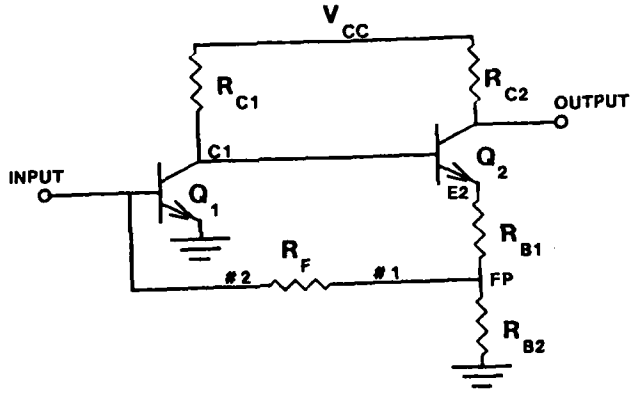


Figure 5 : Feedback Amplifier

The correct interpretation was determined to be <[B1 IN] [RF V1] [E2 Q2] [RF V2] [RB1 V1] [FP RB1] [Q2 VB] [C1 Q1]>. This interpretation selects many values for the current through RF. The assumption [RF V2] indicates that the voltage at the left hand side of RF causes the current to flow through it. There also exists a signal path through Q1 and Q2 to node FP. KVL can be applied to FP, E1 and GROUND to determine the current through RF. Furthermore, the assumption [RF V1] indicates that the voltage at the right hand side of RF causes the current to flow through it. Since all of these arguments agree on the value of the current through RF, they cannot be distinguished by measurements. However, they lead to different analyses of circuit behavior. If the current through RF was deduced as a consequence of KVL with node FP, the circuit contains feedback. The assumption [RF V2] implies the circuit does not contain feedback.

The assumption [RF V1] can be easily eliminated by re-introducing the compatibility criterion: an interpretation may contain only one assumption about any particular component or node.

The notion of interpretation was introduced in order to capture what is meant by a particular point of view of circuit behavior. Defining an interpretation as a restricted collection of assumptions suffers from a number of problems. Measurement interpretations fail to differentiate between whether feedback is present or not, which is something we expect an interpretation to distinguish. An interpretation could also be defined as a set of assumptions which selects a unique value from each cell. This definition of interpretation is not useful since it requires that the rules be totally independent of each other (i.e. nonredundant).

What is desired is a definition of interpretation that falls between these two extremes, differentiating between essential differences yet permitting redundancies to exists. In order to differentiate between essential and inessential differences something different from a set of assumptions must be utilized.

Using assumptions to represent interpretations raises some theoretical questions. The assumptions are part of the mechanism causal reasoning utilizes to analyze circuits. At the causal analysis level, a point of view is established by identifying how each component contributes to the global behavior. These are different concepts. The extrinsic behavior of each device is given by the particular intrinsic rule of the device model that was used in the analysis. This intrinsic rule can be unambiguously indicated by the name of the variable that triggered the device model. This is specified by [<device> <triggering-variable>]. This motivates the definition of another type of interpretation. A *causal* interpretation is a set of local extrinsic specifications which assigns a unique behavior to each device. Note that this definition permits some redundancy since it allows the same input to be triggered by multiple values.

This definition of interpretation is closely related to the original definition. Every application of a KCL-heuristic or KVL-heuristic specifies, in effect, a local extrinsic behavior for that component or node. Therefore every causal interpretation contains an environment as a subset. This was the source of the confusion.

A causal interpretation contains two kinds of local extrinsic descriptions: those which result from the application of heuristics and those that originate from the application of the basic device models. Causal reasoning employed the heuristic local extrinsic descriptions as assumptions. If we want to utilize these assumptions for interpretations, we must augment them by including

the basic extrinsic descriptions. This solves the problem with the amplifier. The deduction that the voltage derived by KVL is used to deduce the current through RF is described by the [RF V]. Since [RF V2] and [RF V] are two different extrinsic descriptions of the same device, they cannot both occur in the same causal interpretation. [RF V1] is eliminated for the same reason. The compatibility criterion has, in effect, been re-introduced in a stronger form.

Using this definition, the correct causal interpretation for the amplifier is: <[B1 IN] [FP RB1] [Q2 VB] [C1 Q1] ; [E2 KVL] [RF V] [RB1 I] [RC1 V] [RC2 I] [RB2 V] [Q1 V]>. The ";" distinguishes the heuristic and basic extrinsic specifications. Note that [E2 Q2], [RF V2], [RF V1] and [RB1 V1] which were present in the correct measurement interpretation are absent. Instead the basic specifications include [E2 KVL], [RF V] and [RB1 I]. The deduction of a voltage at a node from other voltages or directly from device models is specified by [<node> KVL]. Since this can lead to very lengthy interpretations, the convention used here is that this type of extrinsic specification will only be included if some interpretation of the behavior employs a KCL-heuristic at that node.

Causal interpretations are an artifact of the point of view taken in the analysis. For most circuits, it is difficult to determine by measurements which causal interpretation governs circuit behavior. Causal interpretations are a consequence of how the analyzer chooses to explain the circuit behavior, and not of any objective property of the circuit's behavior. In order to determine, by measurements, that feedback is present in the amplifier, more detailed measurements have to be made than just determining whether some quantity is increasing or decreasing.

Three criteria have been identified for interpretations: An interpretation must

(1) select consistent values.

(2) have compatible local extrinsic descriptions.

(3) contain a maximal number of assumptions.

This list of criteria will be extended in the next chapter.

5.6 Computing Interpretations

This section is a short digression concerning the computational issues raised by the criteria. First, for purely efficiency reasons some of the criteria can be applied during the causal analysis. Second, although the criteria indicate which tentative interpretations are to be rejected they provide little insight into how candidate interpretations can be constructed.

Consistency and compatibility are intensional criteria that can be applied to environments individually. Therefore, for efficiency sake QUAL applies the intensional criteria during the causal analysis. For example, whenever some environment is discovered to contain a contradiction all propagations within that environment and any superset environments are stopped. From a theoretical point of view it makes no difference when these criteria are applied. In order to apply the compatibility criterion, the environments of the propagated values are augmented by the local extrinsic descriptions. Thus the environment of each value meets the first two interpretation criteria.

The only way that a new environment can be constructed during the causal analysis is by the application of a heuristic. (The discovery of a new local extrinsic description only extends the environment in which it occurs.) Thus the only cells which need to be examined are those for which a connection heuristic discovered a value. The interpretation constructor examines these cells one at a time. At each iteration step it constructs a new set of partial interpretations by combining the partial interpretations constructed in previous steps with the environments of the values in the current cell. (The iteration starts with a single empty interpretation.) Each partial interpretation is combined with each value by performing a set union on the partial interpretation and the new value's environment. Thus an iteration step starts with n partial interpretations and a cell containing m values, it may discover $n*m$ new partial interpretations. Since an interpretation may select no values from a cell, the next iteration step may begin with $n*(m + 1)$ partial interpretations. In practice this potential exponential increase in partial interpretations never occurs. Most of the newly constructed partial interpretations are contradictory or incompatible.

The correct interpretations are then determined by computing the maximal elements of the resulting tentative interpretations (this could be done using a kind of subset relation). However, most of this computation is avoided by removing the nonmaximal partial interpretations at each step of the iteration. One way QUAL does this is by forcing a partial interpretation to select a value from each cell examined in the iteration (since every partial interpretation which is constructed from it during a step will be a superset of it). As stated this strategy makes the interpretation construction incomplete. For example, a value may not be compatible with a partial interpretation, but may be compatible with some subset of the partial interpretation. QUAL detects these cases and performs simple backtracking.

5.7 The Causal Graph

The causal interpretation distinguishes a unique causal argument for every circuit quantity that is affected by the inputs. The fact that a cell contains a voltage or a current is only important to identify the models which are connected to the cells. Once the topology of interactions is constructed, the details of whether the cell represents a current or a voltage are irrelevant. The causal arguments state how information in one cell affects the information in other cells. Thus the causal interpretation determines the direction of information flow everywhere in the circuit. This information flow is commonly referred to as causal flow.

The flow of information determined by a causal interpretation can be represented by an acyclic digraph. A vertex of this graph represents information contained in a cell, and a directed edge represents the fact that information in the vertex adjacent from the edge contributes to the information in the vertex adjacent to the edge. The *causal graph* represents how the behavior of the individual components contributes to the composite behavior of the circuit.

The requirements of understanding and recognizing circuits demand that such a representation be developed. In order to analyze a circuit the particular details of the implementation of the circuit must be ignored, and the fundamental mechanism by which the circuit achieves its purpose must be identified. The causal graph provides a primitive representation of this mechanism. In order to identify the mechanism the causal graph must be further refined and compared to a library of known mechanisms.

Causal analysis alone rarely identifies a unique causal interpretation for a behavior. Each causal interpretation leads to a different causal graph, only one of which can be correct. A theory of causal graphs is necessary to identify the most reasonable causal interpretation. It also provides a basis by which to perform a differential diagnosis to determine which causal interpretation is most plausible.

The causal graph provides a primitive representation for describing the mechanism by which a circuit achieves its input-output behavior. This behavior is produced by changing input signals causing changes in output signals. Therefore we will only consider that subgraph of the causal graph which describes changing behavior. The vertices of this graph represent changing circuit quantities and the edges in this graph represent the fact that a change in one quantity directly causes a change in another quantity. Since the only uncaused changes are inputs, the inputs are

represented by the only vertices of in-degree zero. Furthermore, since we are only interested in changes that eventually cause changes in outputs, the outputs are represented by the only vertices of out-degree zero.

The causal graph corresponding to the inverting behavior of the DTL-inverter is:

```
DTL-INVERTER-2 <(Q1 ON) [+B D2] [D2 V1] (D2 ON) [+N D1] [D1 V2] (D1 ON);
[RL I] [Q1 V] [R3 V] [R1 V]>
```

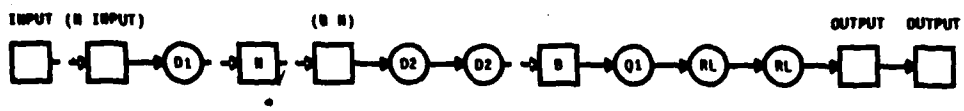


Figure 6 : Causal Graph for DTL-inverter

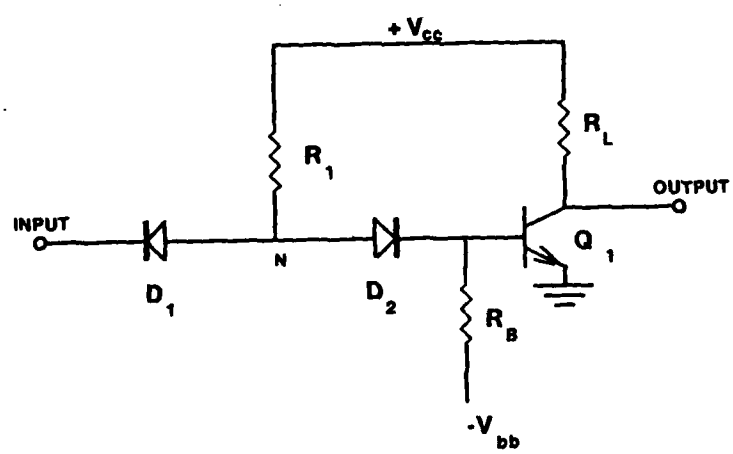


Figure 7 : DTL-inverter

An edge is dashed if the rule that deduced the causal relation between the two quantities made an assumption. Circle vertices represent currents and square vertices represent voltages.

Since the power supply supplies an unchanging voltage, it does not appear in the graph. Although the circuit quantities associated with the bias resistors change, these changes do not causally contribute to the output behavior, and thus do not appear in the graph. This causal graph

is in one-to-one correspondence with the causal argument for the inverter's behavior presented in section 4.8.

In order to avoid cluttering the diagrams, only a minimal amount of annotation is presented on the graphs. If the quantity is a current, it is labeled by the device it is a current of. If the quantity is a voltage with respect to ground it is labeled by the name of the non-ground node. Otherwise the voltage vertex is labeled by its pair of circuit nodes. If there is room, the label is printed inside of the vertex, otherwise it is printed above it. Although this annotation is incomplete, reference to the circuit whose behavior it represents should easily dispel any ambiguities.

Not all causal graphs are straight lines. A causal graph may contain *splits* and *joins*.

Figure 8 : Splits and Joins.

A split occurs at a vertex with out-degree greater than one. Such a vertex represents a quantity that directly causes two other quantities to change, both of which eventually affect the output. A join occurs at a vertex with in-degree greater than one. In this case the change in the quantity is caused by the simultaneous change in all the antecedent quantities. Note that a join does not represent alternative arguments for the same change; all of the antecedents are necessary. Most splits and joins are the direct result of particular circuit mechanisms, but some splits and joins are necessary to account for the difference in number of inputs and outputs.

The term *feedforward* is used to describe the situation when two paths originating at a split combine at a join. Feedforward is rare. The main purpose in examining it is that it is dual to feedback. Incorrect interpretations of feedback behavior invariably lead to the appearance of feedforward. An understanding of valid feedforward provides a method for identifying incorrect interpretations of feedback.

A circuit that exhibits valid feedforward is a simplified complementary-symmetry pair:

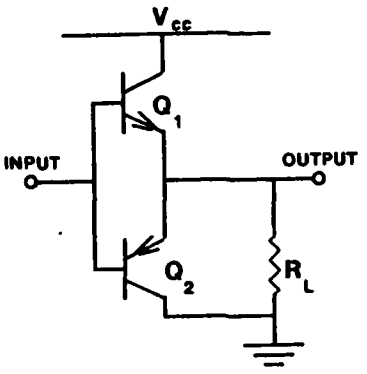


Figure 9 : Unbiased Complementary Pair

In response to an increase at the input, Q1 turns further on, increasing the current out of its emitter. On the other hand, Q2 turns further off, decreasing the current into its emitter. Both of these effects contribute to an increase at the output. This is an example of feedforward. Both of these quantities contribute to the join at the output node. Each of these quantities can also individually cause the output behavior, but under a different interpretation. In any interpretation where Q1 and Q2 both are on, both emitter currents must be included when considering the output node. Since the base-emitter junctions of the transistors are directly connected, QUAL demands that the behavior of one of the transistors dominates the behavior of the other. The following causal graph corresponds to the interpretation where Q1 dominates Q2. The split occurs at the base-emitter voltage.

CS-PAIR-2 : ENVIRONMENT-8 = <[Q1 VB] (Q2 ON) (Q1 ON); [OUTPUT KVL] [RL I] [Q2 V]>

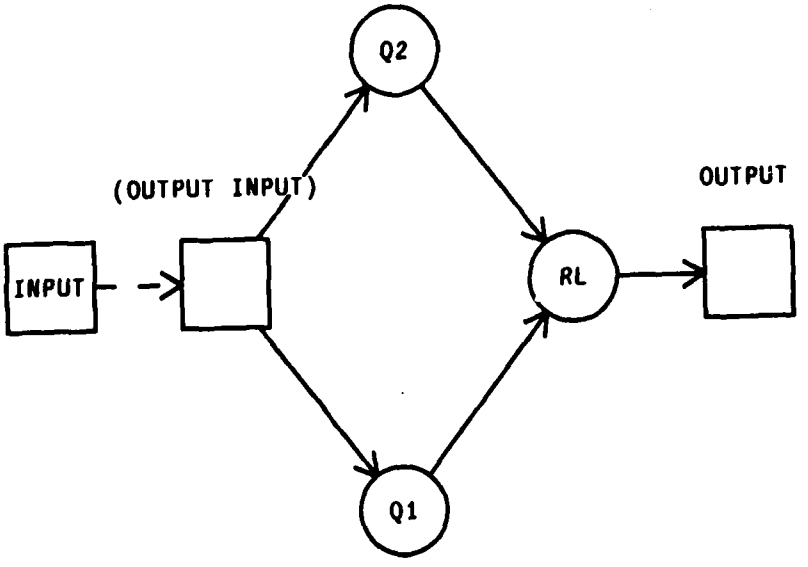


Figure 10 : Causal Graph for Unbiased Complementary Pair

Chapter 6

FEEDBACK

6.1 Feedback is a Global Mechanism

The most important global mechanism by which electrical circuits achieve their behavior is feedback. Feedback is as important and prevalent in circuits as loop constructs are in programming languages. Feedback controls the behavior of a circuit by sampling the output and using this signal to adjust the behavior of the components that are producing the output. The detection and analysis of feedback turns out to be very simple. The notions of assumption, interpretation and causal graph have laid out the framework for this. In order to validate the qualitative feedback theory a bridge needs to be built between it and the classical feedback theory of electrical engineering. The complexity of this chapter originates from the fact that this classical theory is very sophisticated.

The study of feedback impacts recognition in two ways. First, the qualitative theory of feedback constrains the combinations of values that are possible in the causal analysis thereby reducing the number of interpretations that have to be considered. Second, feedback provides a language for describing the mechanism of the circuit at a shallower level of detail for subsequent teleological analysis.

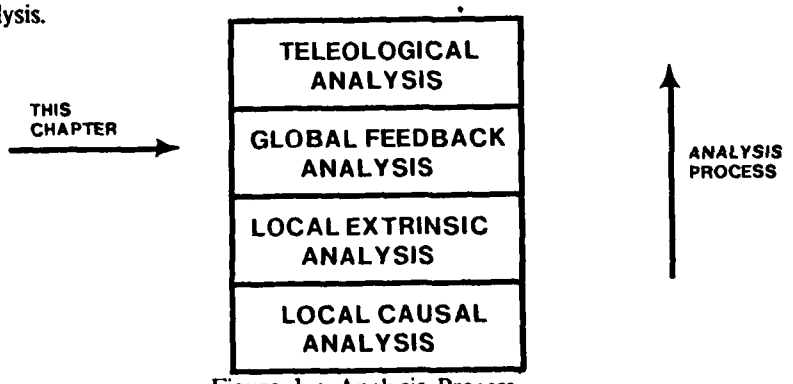


Figure 1 : Analysis Process

6.2 The Feedback Graph

The parameters of individual components cannot be arbitrarily chosen or controlled. For example, the beta of a transistor cannot be precisely specified in the fabrication process, nor can it be precisely controlled when operating since it varies with temperature. Circuits whose behavior needs to be precisely controlled utilize a technique of observing their output signals and using this information to adjust intermediate signals contributing to the output. This is called *feedback*. Feedback is a property of the global behavior of a circuit and cannot be adequately handled by the local rules of the type used for causal analysis. Feedback is dealt with by a meta-strategy which examines causal arguments, but which does not contribute to these causal arguments.

During causal analysis the existence of feedback can be locally detected. The key lies with the causal assumptions. In order for feedback to occur, an input (not the entire circuit's input) signal must be combined with a fraction of the output caused by this input. At this summing point, the original input and the fed back correction combine to produce a modified input. When causal reasoning first considers this summing point only the original input can be known, and thus it cannot propagate past this point without employing a heuristic. The connection heuristic will assume one of the two unknown quantities dominant (i.e. the signal path or the feedback path – locally these are indistinguishable), and continue propagating.

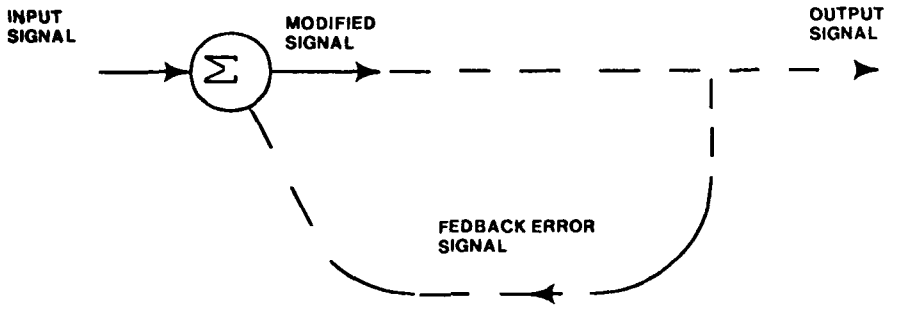


Figure 2 : Feedback

If the connection heuristic assumes that the error signal is not dominant, the modified signal propagates and this eventually determines a value for the error signal. Feedback occurs when a value propagates into a quantity which was assumed to be not dominant, where the propagation

depended on this very same assumption. Since only the KCL-heuristic and KVL-heuristic rules make assumptions, the existence of feedback can be locally detected.

In order to represent feedback, a modified causal graph called the *feedback graph* is used. The feedback graph consists of the basic causal graph of the output, augmented by the causal graphs for the error signals, where extra edges are included to indicate where the error signals propagate into an assumption. Since feedback may not be important to the behavior of the circuit, such edges are dashed in the diagrams.

The correct interpretation of the feedback amplifier's behavior leads to the following feedback graph:

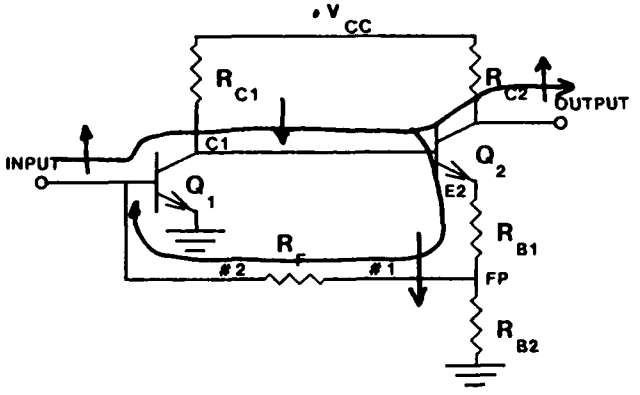


Figure 3 : Feedback Amplifier

The feedback signal is the current flowing from resistor RF into node B1. This current is deduced by applying Ohm's law to the voltage determined by applying KVL to nodes B1, FP and GROUND so both the voltage at B1 and the voltage at FP contribute to the error signal. Therefore the graph must include an edge from vertex B1 to vertex (FP B1) to indicate this dependency, producing a spurious feedback cycle. Further reasoning on these feedback graphs will ignore these spurious cycles.

The feedback graph is a digraph containing cycles. The combination of the causal graphs for the outputs and the feedback graph for each of the occurences feedback is the *mechanism* graph which describes the circuit's complete IQ behavior. The mechanism graph is often too complex

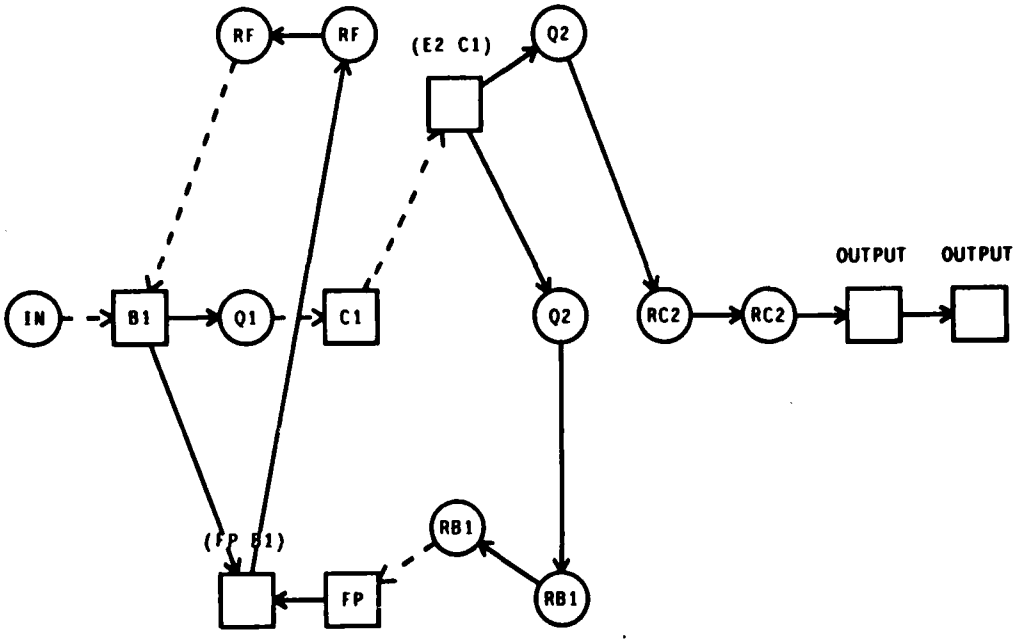


Figure 4 : Feedback Graph for Feedback Amplifier

to fit in a single figure. Many of the examples presented in this chapter are graphs of a single feedback loop.

Each interpretation leads to a particular mechanism graph. This graph describes the feedforward and feedback which the behavior under the interpretation exhibits. This graph has abstracted away a great deal of detail from the original circuit topology. The graph is unilateral while the original topology of constraints are bilateral. The graph can be viewed in terms of the flow of information in the circuit, and not in terms of particular voltages and currents. The next section considers some of the information present in the mechanism graph. There are two reasons for viewing the circuit mechanism as an information flow: the need to describe the behavior at a shallower, more general, level of detail, and the necessity to distinguish between interpretations. Causal reasoning says nothing about how to distinguish between interpretations. In fact, there is no language to distinguish between interpretations other than pointing them out directly. A theory of mechanism graphs will provide a method for how to distinguish between interpretations in a more general way.

6.3 Feedback Configurations

Electrical engineering has developed a considerable amount of theory about feedback. This

section will demonstrate that much of this analytical theory is directly applicable to understanding feedback in incremental qualitative analyses. Once this is shown, we will have gained access to the language electrical engineers use to describe feedback behavior. This language plays a fundamental role in teleological reasoning. The type of feedback exhibited by a circuit is classified according to its sign, the topology of the mechanism graph, and the topology of the underlying circuit. Insight into how the topology of the underlying circuit implements a feedback mechanism is provided by causal reasoning. This section will discuss the different feedback configurations which can be distinguished by consulting the IQ analysis.

Feedback can only occur as the result of a KCL-heuristic or KVL-heuristic. For example, when a current into a node is discovered, the KCL-heuristic sets the voltage at the node as if the rest of the circuit were a positive resistance. Since the KCL-heuristic is usually applied without knowing all of the node currents, the heuristic makes the assumption that the remaining unknown currents are nondominant. If causal reasoning later discovers a value for some of these unknown currents as a consequence of the voltage at the node, feedback is detected. If the fed back current is in the same direction as the other currents, the current into the positive resistance must be even greater and thus the feedback is positive. If the fed back current is opposite to the other currents, the feedback is negative.

On the other hand the KVL-heuristic assumes the voltage at one terminal of a device dominates the voltage at a second terminal of the device. Since this assumption is made when the voltage at the second terminal is unknown, causal reasoning may later discover a voltage at this terminal. If this discovered voltage depends on the voltage at the first terminal, feedback has been detected. Negative feedback is indicated by a fed back voltage equal to the voltage at the first terminal. Positive feedback is indicated by fed back voltage opposite to the voltage at the first terminal.

The vertex of the feedback graph which joins a fed back value with an input value must represent the application of either a KCL-heuristic or a KVL-heuristic. This vertex is called the *comparison point* since it compares an input signal with a fed back signal to produce a composite of the two. The other distinguished vertex of a feedback cycle is the last vertex which causally affects a circuit output. At this vertex a split occurs and one signal continues to the output of the circuit while the other is fed back. This vertex is called the *sampling point.*

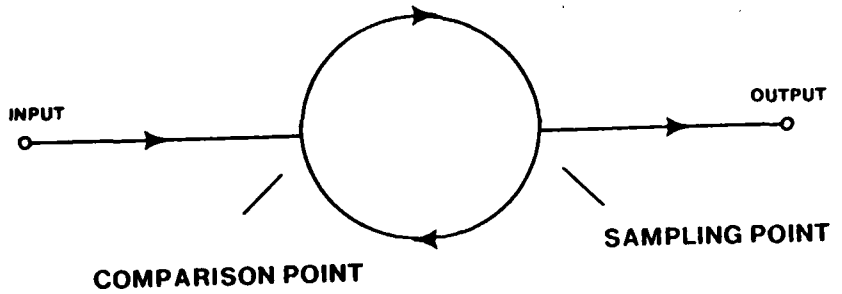


Figure 5 : Sampling Point

The block diagram for a classical feedback system is:

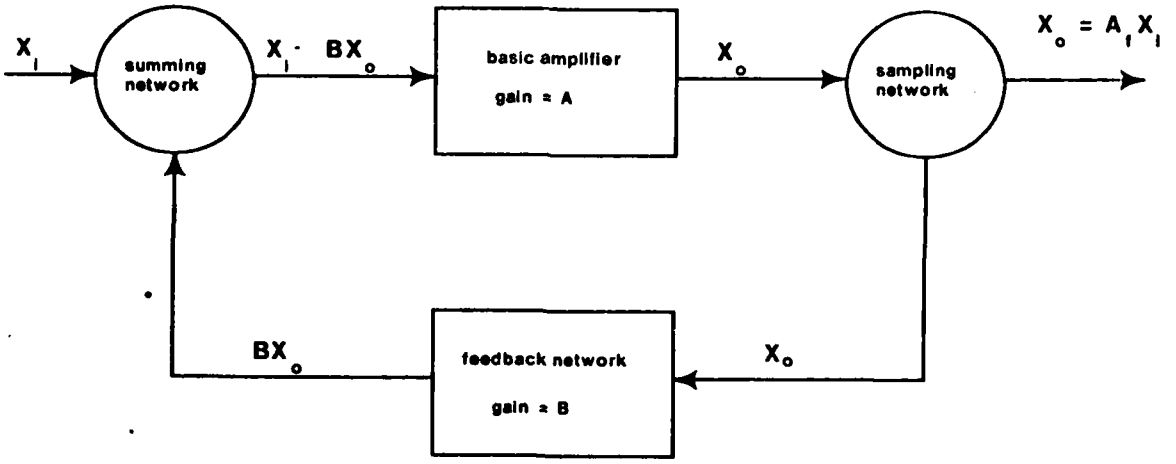


Figure 6 : Block Diagram of Feedback System

The relationship between qualitative feedback and analytical feedback is apparent from the similarity of these two figures.

The method of comparison has direct impact on the input impedance of the amplifier. Negative feedback operates by reducing the difference between the input signal and the fed back signal. (i.e. If the output is high, a high quantity is subtracted from the input signal thus reducing the undesirable high output.) Consequently, the feedback originating from both the KCL-heuristic and the KVL-heuristic tends to move the input voltage and the input current of the *basic* amplifier towards zero. In the case of the KVL-heuristic the input of the *basic* amplifier is in series with the output of the feedback network.

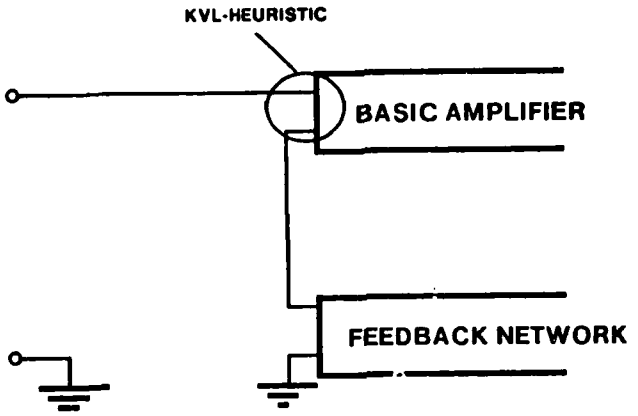


Figure 7 : KVL-heuristic Circuit Topology

Assume the input voltage is fixed. Since the input current of the *basic* amplifier is the same as the input current of the *composite* amplifier, the feedback action of reducing the basic amplifier input current increases the input impedance of the *composite* amplifier. Conversely, when the comparison point is a KCL-heuristic the input of the basic amplifier and the output of the feedback network are connected in parallel.

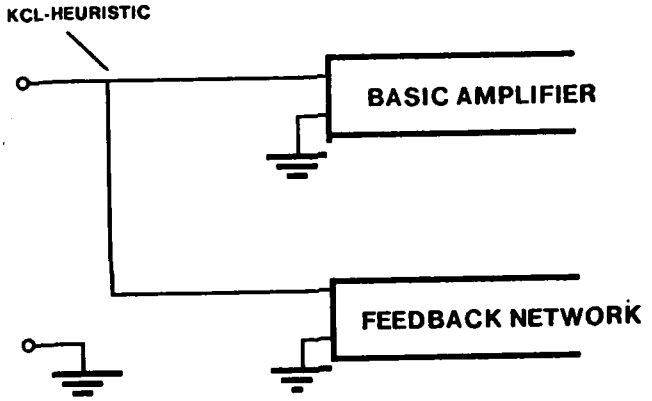


Figure 8 : KCL-heuristic Circuit Topology

Assume the input current is fixed. Since the input voltage of the *basic* amplifier is the same as the input voltage of the *composite* amplifier, the feedback action of reducing the basic amplifier voltage decreases the input impedance of the *composite* amplifier.

The sampling point also distinguishes different feedback configurations. Causal reasoning does not provide as succinct a characterization of the distinctions as it did for the comparison point. The sampling vertex always represents a voltage since currents cannot simultaneously cause

two other quantities to change (i.e. output and error signals). In the classical theory, the two sampling configurations originate from the output of the basic amplifier being in parallel or in series with the feedback network. If the feedback network is in series with the output of the basic amplifier, both of the outgoing edges of the sampling vertex must represent currents. This can only happen if the vertex represents a three-terminal device, and the transistor is the only three-terminal device considered here.

The type of sampling affects the output impedance of the amplifier. The arguments are analogous to the ones used to determine the effect of the comparison type on input impedance. The feedback action tends to stabilize the sampled quantity, thus increasing the output impedance in the series configuration and decreasing the output impedance in the parallel configuration.

There are a variety of different nomenclatures for describing feedback configurations. The terminology used here is from [Gray & Searle 69]. The comparison and sampling can be of either node or loop type. The *loop* type describes the situation when the networks are in series and the *node* type describes the situation when the networks are in parallel. The qualitative behaviors of the different feedback configurations are summarized in the following table.

Configuration	Stabilizes	Input Impedance	Output Impedance
Node-node	Transresistance	Low	Low
Node-loop	Current gain	Low	High
Loop-node	Voltage gain	High	Low
Loop-loop	Transconductance	High	High

Table 1 : Comparison-Sampling Configurations

The type of the feedback configuration also provides direct advice about how to analytically determine the behavior of the circuit. Although not relevant to this research, it is an example of how causal analysis can help programs such as SYN. For example, the following advice can be found in most textbooks on feedback amplifiers. In order to analyze a Loop-loop configuration you should:

(1) Use z parameters to model the two-ports.

(2) Calculate the gain of the feedback network by driving the feedback network with a current and determining the voltage produced into an open circuit.

(3) Calculate feedback loading at amplifier input by open circuiting output feedback node.

(4) Calculate feedback loading at amplifier output by open circuiting input feedback node.

If no vertex of the feedback loop is on a direct path to the output, no sampling point can be identified. Such a *disconnected* feedback loop is uncommon and is usually a consequence of choosing an incorrect interpretation. Very rarely the feedback loop may contain feedback within it. For these loops the notion of sampling point must be redefined to be that last vertex of the loop that either directly affects the output or is a member of another feedback loop which affects the output.

6.4 Sample Feedback Analyses

The mechanism graph of the feedback amplifier is:

```
CE-FEEDBACK-2 : ENVIRONMENT-26 = <[-FP RB1] [Q2 VB] [-C1 Q1] [+B1 IN] (Q2
ON) (Q1 ON); [E2 KVL] [RF V] [RB1 I] [RC1 V] [RC2 I] [RB2 V] [Q1 V]>
Loop:1/1; Sign:-; Comparison:NODE; Samplings: LOOP((E2 C1)) LOOP(B1)
```

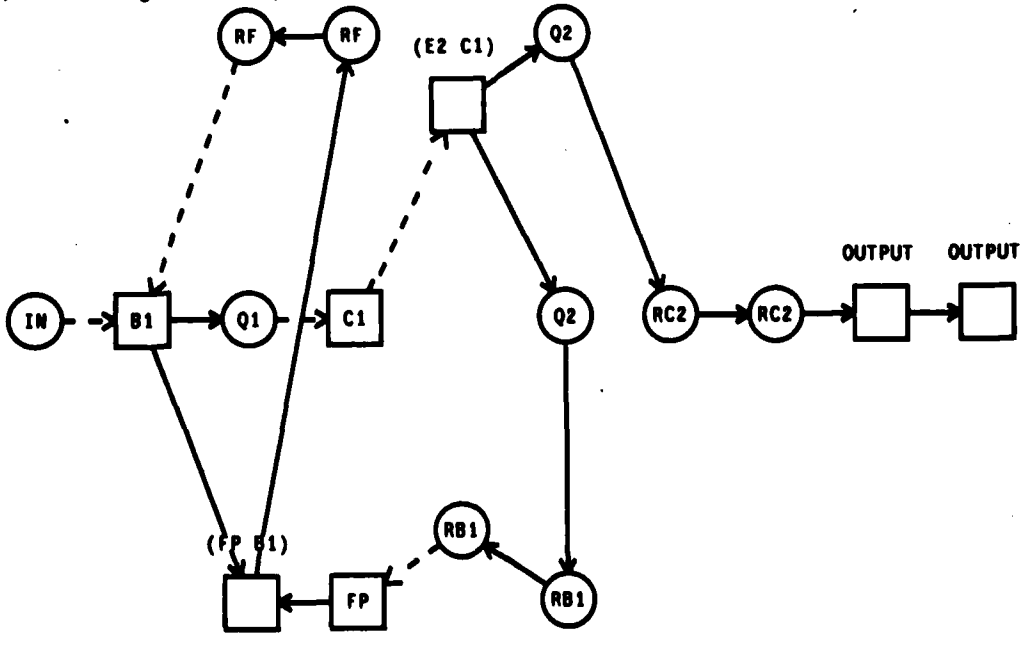


Figure 9 : Mechanism Graph of Figure 10

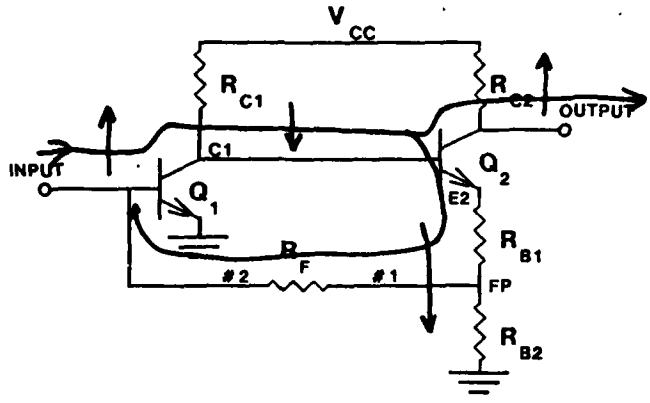


Figure 10 : Feedback Amplifier

Since the fed back signal affects a KCL-heuristic, the comparison is of type node. At the sampling point (E2 C1), two currents are produced, indicating loop sampling. This circuit has low input impedance, high output impedance, and stable current gain.

Another example is the following circuit which illustrates the loop-node configuration:

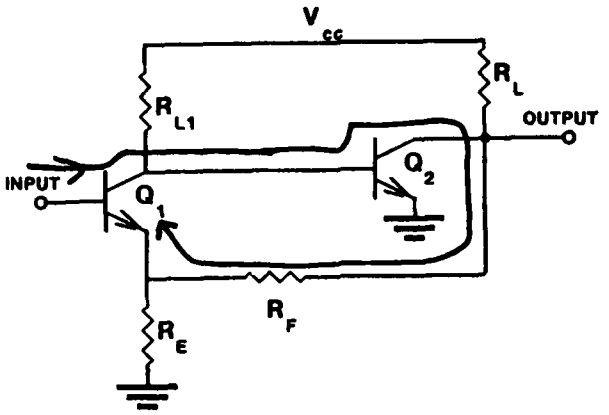


Figure 11 : Loop-Node Feedback Amplifier

Under the correct interpretation, the mechanism graph is:

```
LOOP-NODE-2 : ENVIRONMENT-16 = <[RF V1] [+OUTPUT Q2] [-B2 Q1] [Q1 VB] (Q2
ON) (Q1 ON); [E1 KVL] [RE I] [Q2 V] [RL1 V] [RL V]>
Loop:1/1; Sign:-; Comparison:LOOP; Samplings: NODE(OUTPUT) LOOP((E1 INPUT))
```

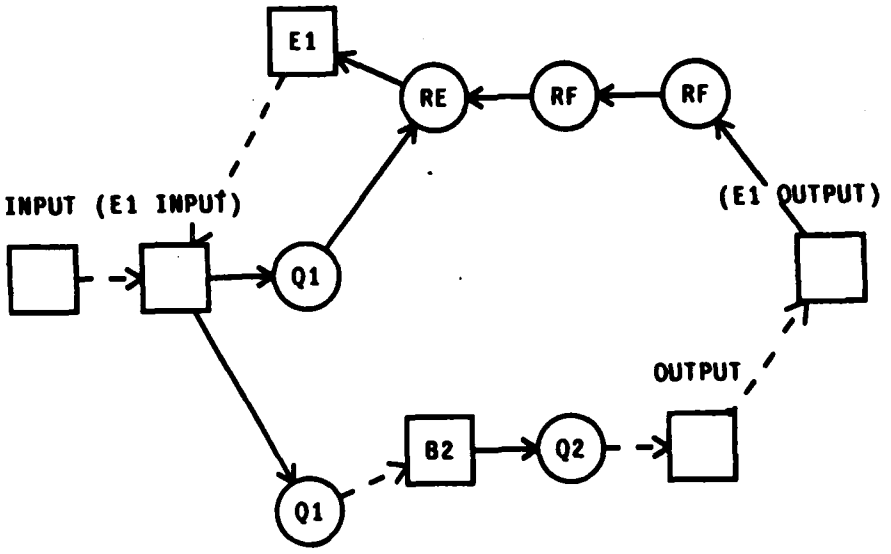


Figure 12 : Mechanism Graph of Figure 11

Since the fed back quantity affects a KVL-heuristic, the comparison is of type loop. The sampling point is the output of the entire amplifier and is of type node. This circuit has high input impedance, low output impedance, and stable voltage gain.

6.5 Local Feedback and Reflections

The comparison point and the sampling point of a feedback loop can be the same. This situation is called *local feedback* and is distinguished from *overall feedback* where the sampling point and the comparison point are different. Since one of the vertices of a local feedback loop is on a path that directly affects an output, the feedback action can have some effect on the circuit's behavior. Unfortunately the techniques developed in the previous section to handle overall feedback do not apply to local feedback. The following simple common-emitter amplifier exhibits local feedback.

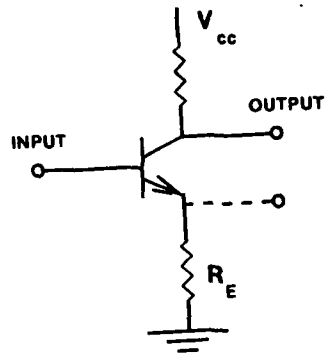


Figure 13 : Common-Emitter Amplifier

```
CE-STAGE-2 : ENVIRONMENT-6 = <[-OUTPUT Q1] [+E Q1] [Q1 VB] (Q1 ON); [RC V]
[RE V]>
Loop:1/1; Sign:-; Comparison:LOOP; Sampling:LOOP
```

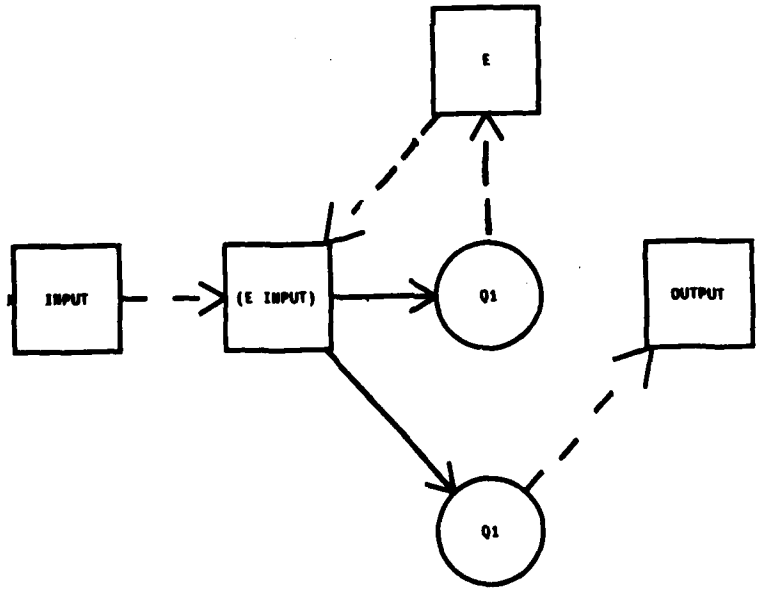


Figure 14 : Mechanism Graph for Figure 13

A rising voltage at the input invokes the KVL-heuristic which applies this voltage to the base of the transistor. The transistor turns on harder, thus increasing its emitter current. Since

this current flows through the emitter resistor, the voltage across the resistor rises, negating some of the effect of the rising input voltage.

In electrical engineering, this type of local feedback is called emitter degeneration. It does not directly fit into the classical feedback pattern since it is impossible to distinguish the input and output terminals of the basic amplifier and feedback network. The circuit can be forced into the classical framework by splitting the emitter current into two quantities. One of these quantities is the input current and the other is the output current. In this way the basic amplifier's missing terminals are created. The feedback can now be identified to be of the loop comparison node sampling type. This technique has no analog in IQ analysis.

Local feedback poses a problem to causal analysis since it cannot be distinguished from the immediate response of the rest of the circuit to the heuristically propagated value. In order for this value to propagate at all, it must affect a succeeding component. This component presents a load, and thus causes a *reflected* voltage or current. Thus local feedback can occur at every point where a connection heuristic is utilized.

The KCL-heuristic assumes the circuit fragment around the node behaves as a positive resistance. Since Ohm's law applies to this fragment, the current flowing through this positive resistance will eventually be deduced, resulting in the detection of feedback.

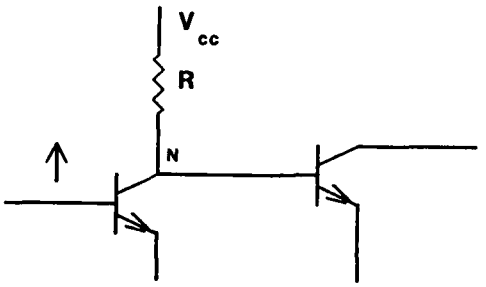


Figure 15 : KCL-heuristic Local Feedback

After the KCL-heuristic asserts a voltage at N, resistor R immediately responds producing negative feedback.

The KVL-heuristic applies a voltage to a nearby component, and IQ rules will propagate this voltage to other components. These components also present loading effects and feedback occurs. Emitter degeneration is an example of a KVL-heuristic reflection.

KCL-heuristics and KVL-heuristics need to be substantiated. An examination of the signals, particularly the reflected signals, around the point at which the heuristic is applied provides the basis for substantiating or rejecting the application.

The currents into a node at which a KCL-heuristic has been applied can be divided into three categories:

(1) currents which the KCL-heuristic employed in its deduction

(2) currents which were deduced as a consequence of the currents in the previous category (usually via the node voltage)

(3) currents which the KCL-heuristic chose to ignore.

The second category can be further broken down into reflected values and values resulting from valid feedback loops. A reflected value is defined as a fed back value which requires at most one more assumption than the original node voltage, or a fed back value which utilizes at most one component in its feedback loop.

Every KCL-heuristic must be substantiated by a negative reflection. (A positive reflection indicates the presence of a negative resistance.) A reflection is not always an instance of local feedback since the single component on the feedback cycle may be on a path to the output. Not every local feedback path is necessarily a reflection. Unfortunately, in the more common instance where the reflection is a consequence of a local feedback loop, the substantiating and local feedback effects cannot be distinguished.

As the analysis proceeds, most of the currents into the node will be deduced. Whenever all but one of these currents are known, KCL may be able to deduce the final unknown current. This current is treated as a negative reflection since it forces the circuit fragment to behave as a positive resistance. If the circuit fragment does not behave as a positive resistance, the resulting contradiction will rule out the heuristic assumption.

While the KCL-heuristic needs to be substantiated by the presence of positive evidence, the KVL-heuristic is substantiated by the absence of negative evidence. A KVL-heuristic is substantiated when the voltage at the assumed nondominant terminal is determined to be nondominant. The presence of feedback is detected when the voltage at the nondominant terminal is determined as a consequence of the voltage at the assumed dominant terminal. This fed back value can either be a consequence of a reflection, a local feedback loop, or an overall feedback loop. Any of these substantiate the KVL-heuristic. A positive feedback reflection is evidence that the circuit contains

a negative resistance, and does not indicate that the heuristic should be rejected. If a voltage can be deduced at the nondominant terminal, which is independent of the the voltage at the dominant terminal, substantiation depends on the precise values of the two voltages. If the voltages are different, the heuristic is substantiated. However, if they are of the same value, the heuristic must be regarded with a high degree of suspicion. Unlike the case with the KCL-heuristic, failure to substantiate a KVL-heuristic is not sufficient to rule it out. If the analysis could have deduced a different voltage directly across the component itself, the contradiction mechanism would have automatically ruled out the heuristic. The remaining difficult case is where both terminals of a device have independently derivable voltages which are of the same sign, but no clear dominance can be established. As a consequence, the analysis must entertain both possibilities.

6.6 Non Signal-Processing Feedback Circuits

The previous discussions focused on the role of feedback in the signal-processing behavior of simple amplifiers. Many circuits which exhibit feedback cannot be usefully viewed as signal-processors of this type. Feedback is sometimes used to prevent an input from affecting an output. This occurs in protection circuits which restrict the currents and voltages inside the circuit to safe levels. Some feedback circuits have no inputs or outputs in the usual sense, and such circuits are unanalyzable by the techniques discussed so far. For example, a power-supply has no inputs and is specifically designed to have no incremental output.

One method to limit the output current of an amplifier is to let the current flow through a resistor which develops a voltage-drop across it which is then sensed by a transistor which feeds back its collector current.

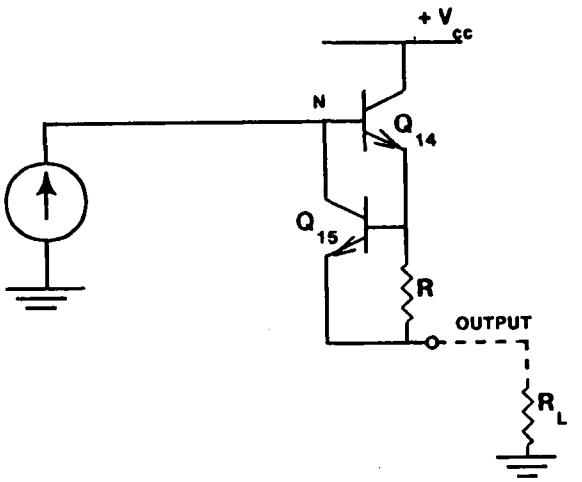


Figure 16 : Current Limiter

The graph for the feedback action is:

```
PROT-2 : ENVIRONMENT-13 = <[-OUTPUT Q15 R] [R V1] [-B15 Q14] [Q14 VB] [-
B14 I-IN] (Q14 ON) (Q15 ON); [Q15 V]>
Loop:1/1; Sign:-; Comparison:NODE; Sampling:LOOP
```

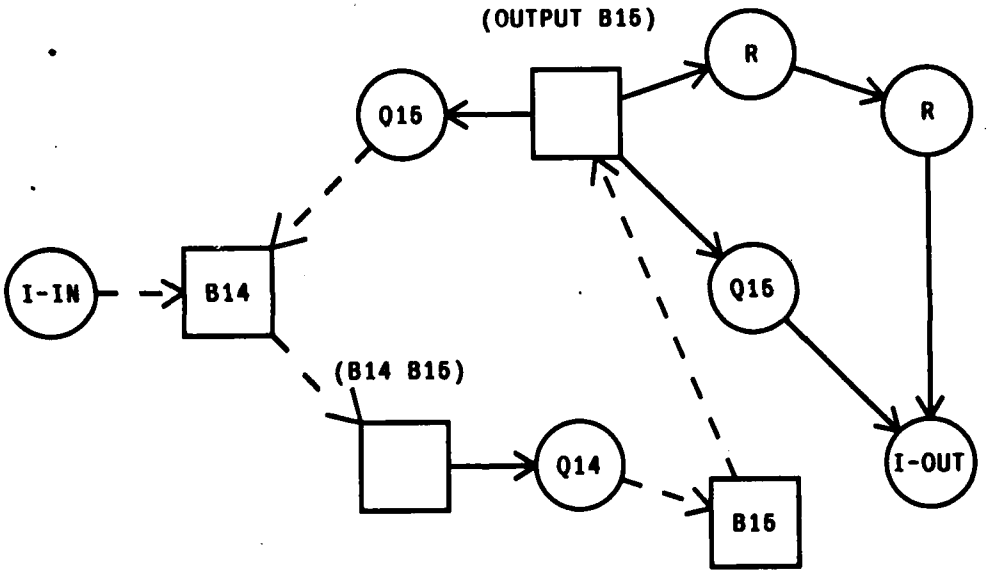


Figure 17 : Mechanism Graph for Current Limiter

The effectiveness of this circuit depends on the exponential behavior of Q15's emitter junction, and this cannot be captured by the IQ rules. Causal analysis identifies that the circuit contains

an active device in its feedback loop and that this feedback stabilizes output current. Both of these features suggest, but don't prove, that the circuit could behave as a protection device.

The purpose of some circuits is to provide a constant unchanging output in the face of changing inputs. The following is an example of such a circuit. This circuit attempts to provide a constant output current even if the voltage applied to it varies.

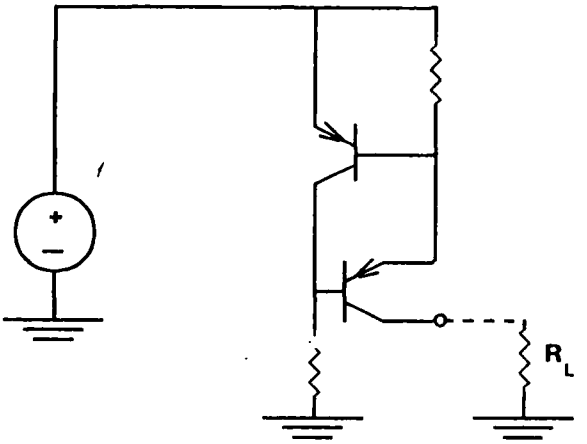


Figure 18 : Constant Current Source

Causal analysis finds that the output current varies with applied input voltage. IQ analysis fails to explain circuits with no incremental output because it is oriented towards signal-processing circuits. It fails to account for the fact that negative feedback may effectively eliminate all incremental output. This problem is partially dealt with by asserting a zero signal at the basic amplifier input point when negative feedback is detected. For example, when negative feedback is detected in the current limiter, a zero voltage is asserted at node N. This assumption is labeled [0N I-IN]. Since this voltage does not propagate, the current limiter has zero output under this interpretation.

This patch fails to explain the constant current source (CCS). The only negative feedback the CCS contains is local and cannot be distinguished from reflections. Even if the reflections were regarded as valid negative feedback, the above technique still fails. The difficulty is that the circuit is an amplifier of very small, but non-zero, gain. (The relevant gain for this amplifier is its transconductance — the ratio of the incremental output current and the incremental input voltage.) A judicious choice of parameter values makes this gain negligible. The IQ models are too coarse to make this argument, and yet are detailed enough to determine that the gain of

the CCS is non-zero. Causal analysis can only determine that the circuit possibly contains local feedback which tends to lower its gain.

Note that if this current source is part of a larger amplifier, it presents no problem since it has no input terminal other than the main power supply. Hence causal analysis correctly determines that the current source is part of the bias circuitry.

A power supply has no input in the usual sense (see figure 19). This circuit is of the loop sampling node comparison configuration. The amplifier presents a high-impedance to the reference battery, ensuring long battery life. It has a low output impedance, as is desirable for a voltage source. This configuration also tends to stabilize voltage gain, thus the output voltage will be a constant multiple of the reference voltage.

Since this circuit has no input, causal analysis cannot detect the above behavior. When the circuit is functioning, the amplifier has no incremental input. At best, the output current can be treated as an input with the output voltage as the output. This leads to a local feedback loop centered at the output.

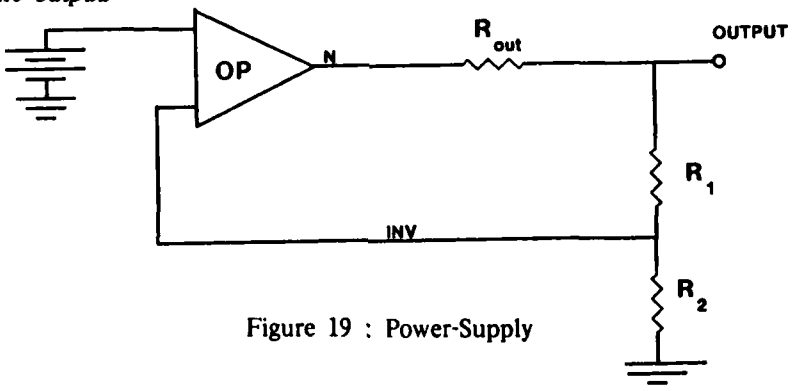


Figure 19 : Power-Supply

From this graph (figure 20), the loop-node configuration cannot be discerned. The local feedback shows that the feedback action tends to stabilize the output voltage and lower the output impedance of the circuit. This is what we expect from a power-supply.

All three of these circuits present a similar problem to causal analysis. None of these circuits' ultimate purpose is amplification. Modifying causal analysis to deal with these other circuit types does not address the central problem. The protection circuit and the constant current source can, quite correctly, be considered to be amplifiers. By calling a circuit a *protection circuit* we are saying it serves a particular role in a larger framework. Whether or not a circuit fulfills its

```
SERIES-REG-2 : ENVIRONMENT-8 = <[-INV R1] [R1 V1] [-OUTPUT TERM1] (OP LIN);
[N KVL] [ROUT V] [R2 V] [OP V]>
Loop:1/1; Sign:-; Comparison:NODE; Sampling:NODE
```

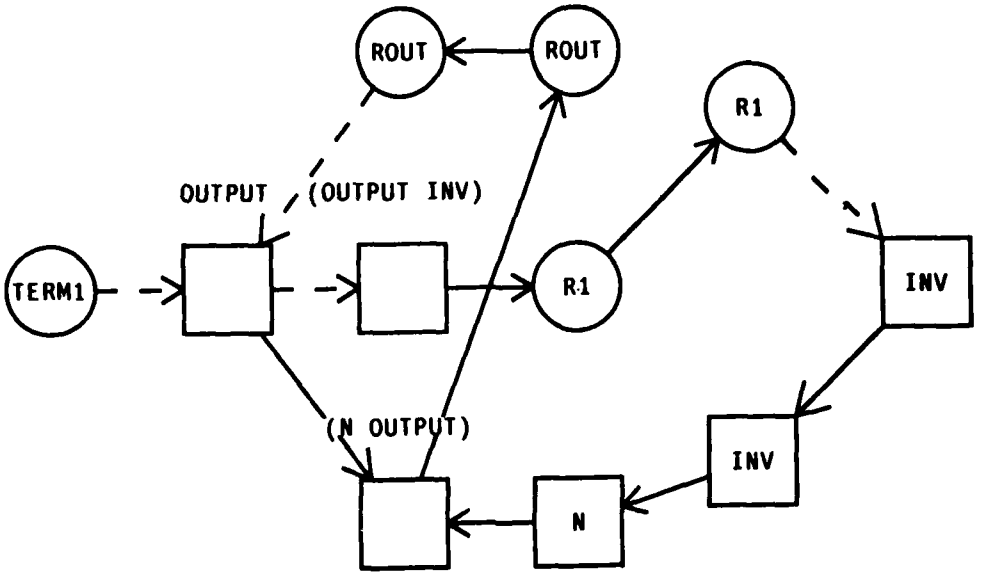


Figure 20 : Local Feedback Graph of Power Supply

role in the larger framework cannot be determined by causal analysis. In all three cases causal analysis finds a mechanism which is consistent with the circuit's global role. This is a weaker kind of rationalization than those considered in chapter 4. An interpretation can be a rationalization since it does not rule out other behaviors. This new type of rationalization says that a particular interpretation may fulfill a given role, but it need not, and the same interpretation may fulfill other roles.

In the case of the power-supply, the knowledge that the circuit is a power-supply and that it is implemented with a loop-node configuration can be used to recognize the circuit. For example, the local feedback loop at the output can be examined to find a KVL-heuristic point at which unexplained components are attached. Knowledge of the global role of the circuit enables causal reasoning to test whether it fulfills its purpose. For example, once the amplifier has been identified in this way, its input can be temporarily disconnected and a test signal applied. Causal analysis can then determine the configuration and properties of the circuit and confirm the hypotheses.

6.7 Regenerative Circuits

While negative feedback tends to stabilize circuit behavior, positive feedback tends to destabilize it. The stabilizing action of negative feedback motivated us to examine the different configurations which stabilized particular circuit quantities. The destabilizing action of positive feedback is utilized in very different kinds of circuits where the particular feedback configurations, while identifiable, play almost no role. Circuits such as triggers, memory elements and oscillators rely on positive feedback to produce behavior which varies with time. Causal analysis does not employ a very sophisticated notion of time and therefore cannot deal with the behavior of these circuits to the same depth as those circuits which exhibit negative feedback. All of the circuits discussed so far have one stable state. Whatever state an input may drive these circuits into, they return to a unique stable state when this input is removed. Positive feedback enables the existence of more than one stable circuit state. Engineers usually do not analyze these circuits directly, but employ rules of thumb to avoid these difficulties. This section will briefly discuss some of the ways causal analysis can deal with these circuits with these same rules of thumb.

Consider an amplifier which feeds back a voltage which it compares to the input voltage (loop comparison).

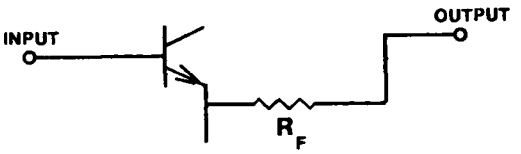


Figure 21 : Positive Feedback Amplifier Input

When the feedback path RF is disconnected, there will be some value of the input which produces a voltage at the output equal to the voltage at the emitter of the input transistor. At that operating point, RF can be reconnected without disturbing anything. Suppose a perturbation is applied to the input. If the perturbation is positive, the feedback action will add a fraction of the output to the input thereby producing the effect of a larger perturbation. If the perturbation is negative, the feedback action will reduce the quantity it was adding to the input thereby producing the effect of a larger negative perturbation. This action will instantly drive the operating point into

a neighboring state which does not exhibit positive feedback (this assumes the gain is sufficiently high). A state that exhibits this type of behavior is termed *meta-stable.*

The circuit has only one operating point within the meta-stable state that is at equilibrium and this equilibrium point is unstable since any small perturbation will drive it away from equilibrium. This action will keep the two neighboring states of the meta-stable apart, thereby forming two stable states.

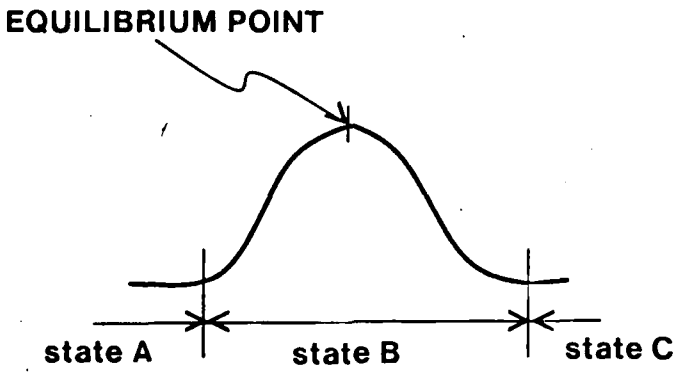


Figure 22 : Meta-Stable point

An analogous physical situation arises in the case of a ball perched on the crest of a hill. Any small movement will result in the ball rolling down one side or the other of the hill.

Circuits which have two stable states are called *bistable.* Since their output behavior depends on past inputs as well as the current inputs, bistable circuits have memory. Most of the circuits that employ positive feedback exploit this property. For example, bistable circuits form the basic memory element of digital computers. To successfully analyze these circuits these past inputs must be taken into account, and this requires reasoning about time. Electrical engineers employ a number of rules of thumb to guide the analysis of bistable circuits, and QUAL is capable of utilizing these. The results of these rules of thumb are then used to guide QUAL in its causal reasoning about time.

The notion that a circuit has a stable state has not arisen before. QUAL does not have a sufficiently powerful quiescent analysis procedure to determine which stable state is correct. However, the strategies it uses for detecting negative feedback also detect positive feedback and thus it can detect meta-stable states. As an example consider the following bistable circuit.

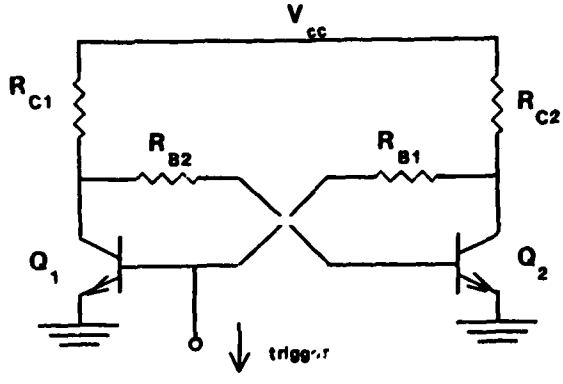


Figure 23 : Bistable Circuit

Under the correct interpretation the state transition diagram of this circuit in response to a falling trigger applied to the base of Q1 is:

```
BISTABLE-2 : Q1 Q2
```

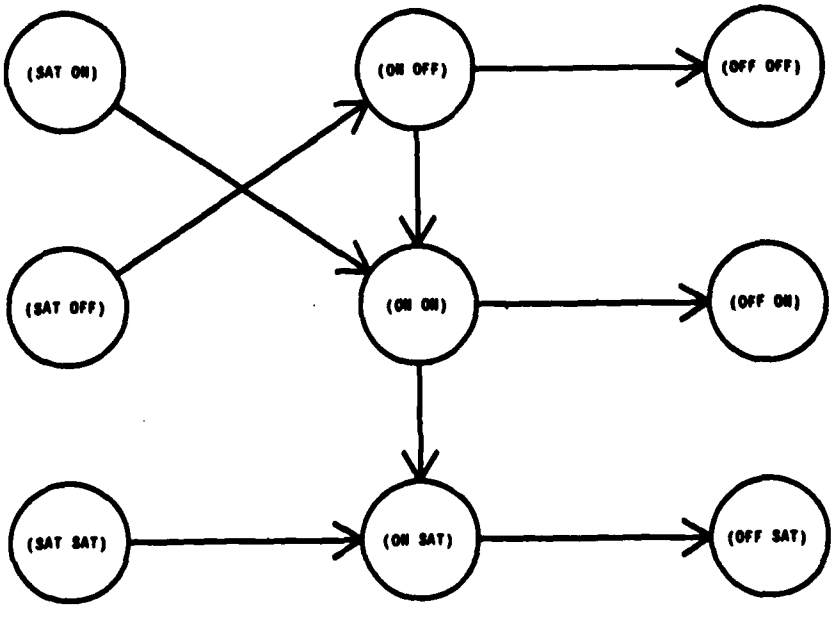


Figure 24 : Transition Diagram for Bistable Circuit

The only meta-stable state is (ON ON). There are interpretations of this state that do not exhibit

positive feedback. The correct interpretation provides a rationalization for the existence of bistable behavior.

Bistable circuits form the foundation for a number of other useful circuits. The introduction of capacitance and inductance generates other possibilities. The addition of capacitance allows the circuit to have different numbers of ac and dc stable states. The three basic types are enumerated in following table.

Circuit Type	Number of dc stable states
Bistable(dc)	2
Monostable	1
Astable	0

Table 2 : Types of Bistable Circuits

A bistable(dc) circuit is indicated by the presence of positive feedback. QUAL can also check the rules of thumb that electrical engineers utilize for determining whether a circuit is monostable or astable.

A monostable circuit produces a pulse of fixed length in response to a trigger. The monostable is customarily constructed from a bistable circuit in which one of the coupling resistors has been replaced with a capacitor. The rules of thumb for monostable behavior are:

(1) With capacitors removed, the circuit must have only one stable state. At least one of the transistors must be out of the active region in this state.

(2) The coupling network must be such that both transistors can be in the active region at the same time.

(3) The positive feedback at ac must be greater than unity.

Condition (1) can be checked by the absence of positive feedback when capacitors are modeled open. Condition (3) can be checked by the presence of positive feedback when the capacitors are modeled shorted. The states of the transition diagram can be examined to check whether condition (1) holds.

An astable circuit has no stable states and thus continuously oscillates between meta-stable states. The rules of thumb for astable behavior are:

(1) With capacitors removed, the circuit must have one stable state in which both of the transistors

are in their active region.

(2) The positive feedback at ac (in the stable dc state) must be greater than unity.

Condition (1) can be checked by the absence of positive feedback when capacitors are modeled open. Condition (2) can be checked by the presence of positive feedback when the capacitors are modeled shorted.

QUAL can also perform a causal analysis directly on the circuit utilizing the time-flow capacitor model. Consider the following monostable circuit:

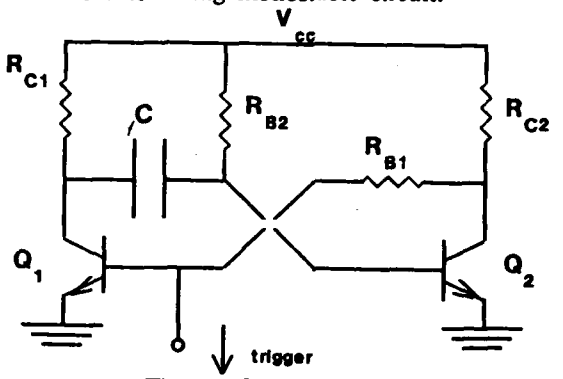


Figure 25 : Monostable Circuit

Both monostable and astable behavior require the presence of capacitors. Astable behavior can be detected by the presence of loops in the state diagram. Monostable behavior can be detected by the presence of a path which returns to its start state in response to a trigger. The complexity of this diagram (figure 26) illustrates the problems which result from the lack of a time-domain signal analysis. The rules of thumb could be utilized to prune this transition diagram down to the indicated path. The passage of time produces too many possibilities for causal analysis to effectively deal with. In order to analyze these circuits their purpose needs to be known such that the correct stable state and appropriate capacitor model can be chosen.

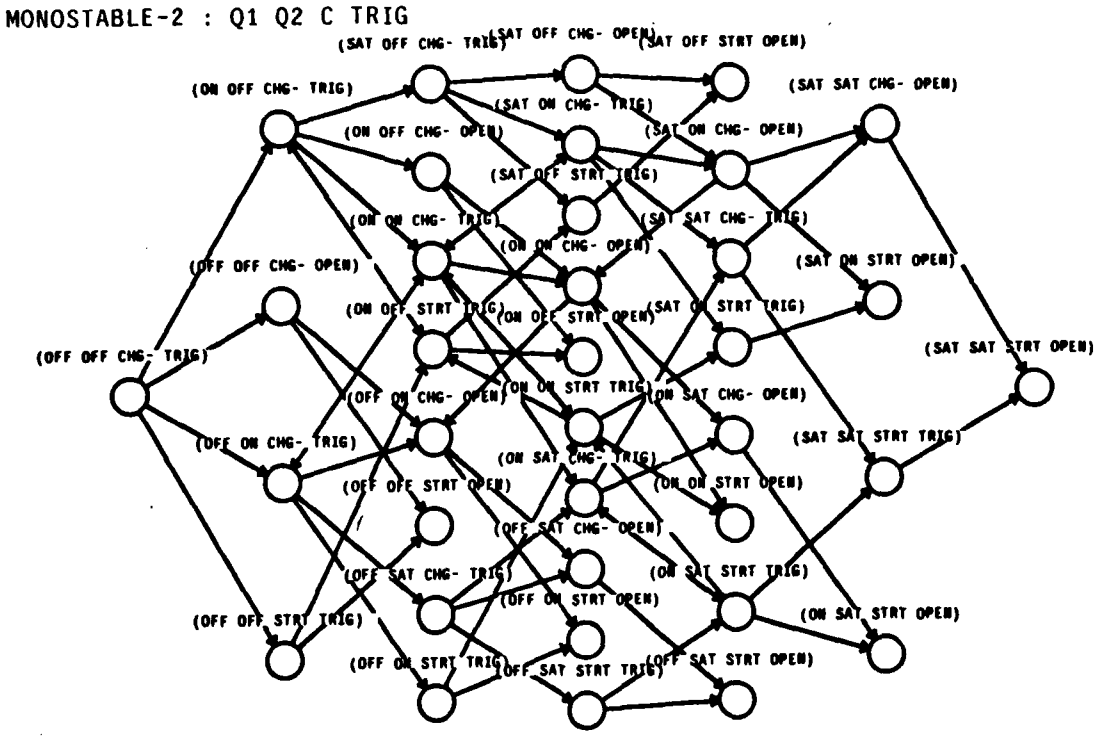


Figure 26 : State Transition Diagram For Monostable

6.8 The Feedback Analysis Process

The global feedback analysis has some impact on identifying the correct interpretation, but its primary role is to generate a description of the global behavior which can be used in ensuing teleological analysis.

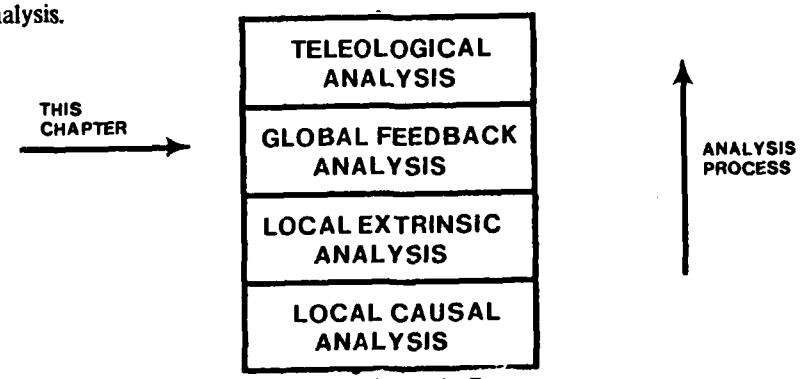


Figure 27 : Analysis Process

The distinctions developed in the previous sections permit the statement of two more interpretation

criteria. These five criteria are applied to the causal analysis of CE-FEEDBACK and this application will suggest what role teleological analysis must play in recognition.

One of the new criteria is that each occurance of a connection heuristic must be substantiated by a reflection. A connection heuristic makes the assumption that a circuit fragment is acting as a positive resistance, and if that assumption is true the circuit fragment must respond with a reflected current of the correct sign. Therefore any interpretation which selects a connection heuristic must also select values which substantiate it. This criterion does not test whether the assumption is valid, but rather whether deductions which ensue from making the assumption at least support it.

The final interpretation criterion is that all *uncaused* currents must be explained. A current is uncaused when KCL constrains it to flow through a device, but the device rule does not predict it. For example, KCL at a collector may force a collector current; if this collector current is not eventually explained by a signal at the emitter junction, the interpretation must be rejected. For elementary circuits these two interpretation criteria are not necessary, but they play a major role in unraveling the intertwined feedback loops which will be discussed in the following two chapters. Experience with QUAL indicates that the number of interpretations is never more than two times the number of active devices. Usually it is the case that the ratio of interpretations to active devices decreases as the size of the circuit increases.

Five criteria have been identified for interpretations: An interpretation must

(1) select consistent values.

(2) have compatible local extrinsic descriptions.

(3) contain a maximal number of assumptions.

(4) substantiate the connection heuristics.

(5) have no uncaused quantities.

The last two criteria are extensional as they require that the causal analysis be finished before they can be applied. They also pose computational problems when applying the maximality criterion. Unlike as is the case with the consistency and compatibility criteria, a subset of an interpretation may not meet the final two criteria. Thus the backtracker (see 5.6) may have to backtrack a number of values. For example, if a tentative interpretation contains an unsubstantiated connection heuristic, that heuristic must be removed before that interpretation can be reconsidered, but an arbitrary number of values and assumptions may have depended on that heuristic.

Analysis of the (ON ON) state of CE-FEEDBACK results in four interpretations. The correct interpretation can be seen in figures 9 and 10. The first incorrect interpretation analyzes feedback as feedforward.

```
CE-FEEDBACK-2 : ENVIRONMENT-19 = <[-C1 Q1] [+E2 RB1] [RB1 V2] [+FP RF] [RF
V2] [+B1 IN] (Q2 ON) (Q1 ON); [Q2 V] [RC1 V] [RC2 I] [RB2 V] [Q1 V]>
```

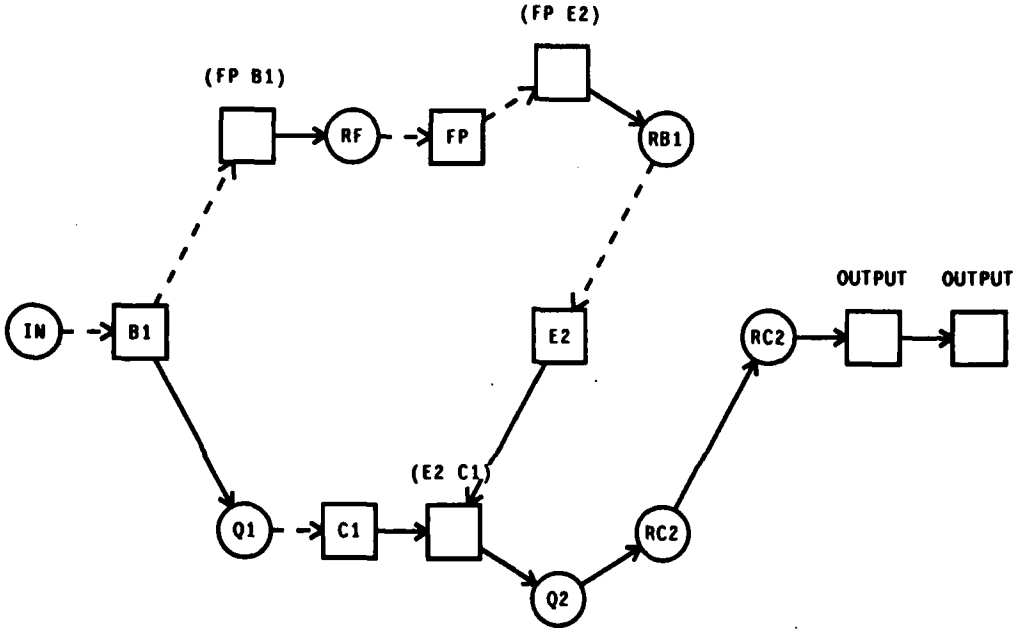


Figure 28 : Feedback as Feedforward

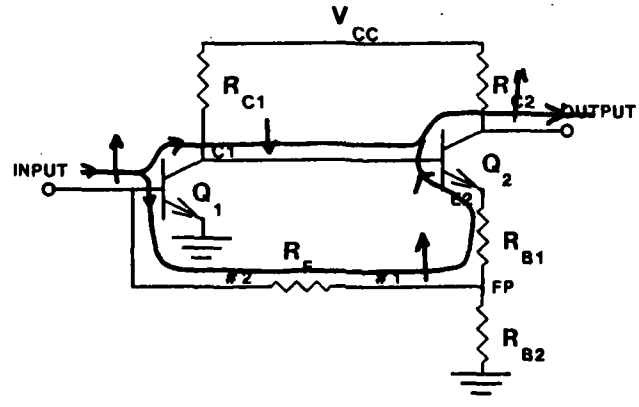


Figure 29 : CE-FEEDBACK

In the second incorrect interpretation the signal goes backwards up the feedback path.

```
CE-FEEDBACK-2 : ENVIRONMENT-25 = <[+C1 Q2] [Q2 VE] [+E2 RB1] [RB1 V2] [+FP
RF] [RF V2] [+B1 IN] (Q2 ON) (Q1 ON); [RC1 V] [RC2 I] [RB2 V] [Q1 V]>
```

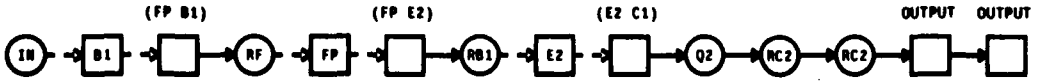


Figure 30 : Unity Gain

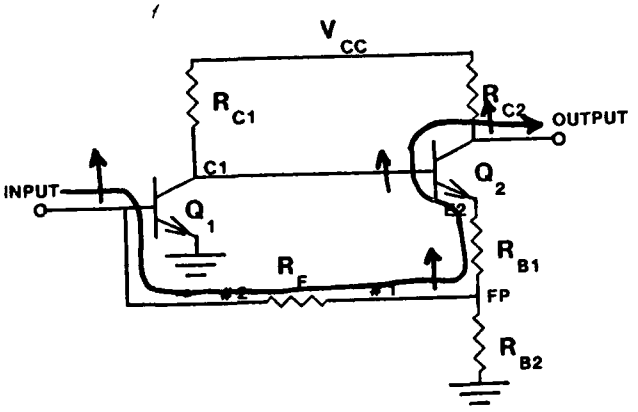


Figure 31 : CE-FEEDBACK

The final incorrect interpretation completely ignores the feedback action.

```
CE-FEEDBACK-2 : ENVIRONMENT-39 = <[-E2 Q2] [Q2 VB] [-C1 Q1] [+FP RF] [RF
V2] [+B1 IN] (Q2 ON) (Q1 ON); [RB1 I] [RC1 V] [RC2 I] [RB2 V] [Q1 V]>
```

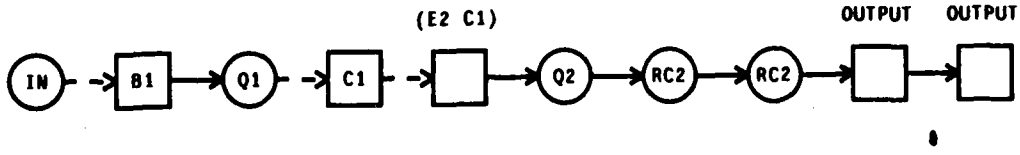


Figure 32 : Feedbackless

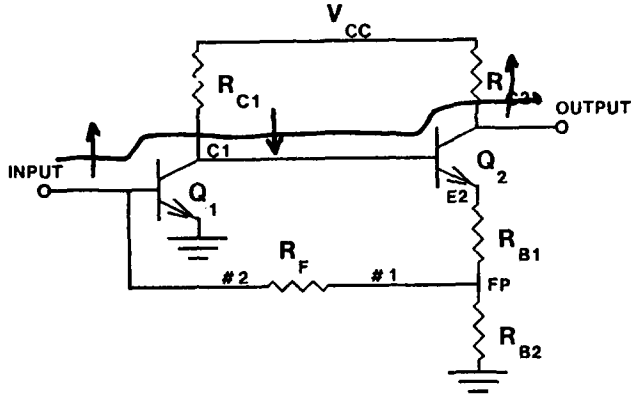


Figure 33 : CE-FEEDBACK

To anyone who knows even a little about circuits these interpretations are obviously nonsensical. The first interpretation (feedforward) feeds forward a signal that is orders of magnitude smaller than the signal it is added to. The smaller signal has no effect and therefore the feedforward through RF serves no purpose. The second interpretation (unity gain) has no gain and assigns no purpose to Q1 or RC1. The third interpretation (feedbackless) assigns no purpose to RF. All of these arguments are based on reasoning about purpose. The next chapter will discuss teleological reasoning capable of making the above arguments and thus identifying a unique interpretation of the circuits behavior.

6.9 Feedback and Constraint

From the constraint point of view, causes for changes cannot be determined since a change at one point affects changes everywhere else. The causal point of view imposes a temporally ordered flow of causality on these changes in which actions cannot affect their earlier predecessors. The ability to extract a causal behavior comes from employing simpler models and connection heuristics. There are some special circumstances where causality must be violated in which a quantity affects one of its predecessors. This is feedback. One way of thinking of the inherent simultaneity present in the constraint point of view is that of feedback being present everywhere. Thus, what causal reasoning really achieves is the determination of which of these feedback loops is central to the circuit's behavior, and how its feedback action achieves this behavior.

Viewed from this perspective, the connection heuristic's sole purpose is to handle simultaneity. These same simultaneities must also arise in algebraic analysis. One of the serious drawbacks of SYN lies with its inability to choose good places to introduce anonymous objects (see section 2.5). The connection heuristics utilized in the correct interpretation indicate places where simultaneity must to be broken. In order for SYN to solve the circuit, it must introduce an anonymous object in loops involving these quantities. The causal interpretation suggests where to introduce the anonymous objects. Furthermore, these variables have a role in the circuit's causal behavior and have significance for the user of SYN.

Chapter 7

TELEOLOGY

7.1 The Teleological Perspective

Circuits are devices designed and manufactured to achieve specific functions. Since these devices have to be conveniently designed, efficiently manufactured and easily maintained, the designer attempts to make his circuits as simple as possible. These desiderata dictate that every component must contribute in some way to the ultimate purpose of the device. For designed artifacts, every component can be related to the ultimate function of the device. This is the *teleological* perspective.

The important result of this chapter is that just the knowledge that a device has some purpose is sufficient to determine the correct interpretation. It is usually not necessary to know what this purpose is.

This chapter commences with an examination of the general role teleology plays in the understanding of physical systems. It contrasts insights provided by teleology to those provided by other methods of gaining the same information. Teleology plays a key role in recognition by providing a method by which to evaluate tentative interpretations. The mechanism graph provides the starting point for hierarchical recognition. As an example, a recognizer is developed for simple amplifiers. The final sections present examples of QUAL's explanation capabilities.

7.2 Designed Artifacts and Natural Systems

It is a common human endeavor to identify purposes for events in nature. The understanding sought for is often very difficult to achieve. The purpose of the heart is to pump blood, but why do humans have color vision? Why does the earth have two tides? If natural systems have a teleology, man is only partially aware of it. In some fields such as the fine arts or architecture the creator may have some idea of the teleology of his creations, but other people may not be able to identify his teleology or may suppose a different one. For example, different

architects will have different opinions about how a building should be laid out. In fields where the artifacts are judged solely by function and efficiency, one sees much more unanimity. Useful mechanical or electrical inventions are quickly and universally adopted. Most engineers will agree that a particular design is a good one. (When engineers differ it is usually because there is some freedom in the functional specification of the purpose of the device.) Moreover, they agree on the purpose of every component. Although teleology plays a role in understanding most domains, it is particularly prevalent in electronics. Teleology plays such a fundamental role in understanding circuits, that electrical engineering has agreed upon a language for discussing it, which makes electronics a very rich domain for exploring this issue.

There is an interesting difference in how people go about investigating natural systems versus designed artifacts. In physics one thinks of discovering the laws of motion. In electronics one asks what role a capacitor plays in an oscillator. Natural systems are investigated by making distinctions and inventing basic laws which, if true, uniquely predict that the observed phenomenon must occur. For example, Galileo made the distinction between force and momentum. Newton established laws based on these distinctions that enabled him to predict that the moon revolved around the earth in the way it does. The situation is radically different for devices made by man for particular purposes. The single fact that a device has a purpose often tells you a great deal. Every component in the circuit must have a purpose and therefore any analysis of the circuit that does not explain every component must be regarded with some suspicion. Furthermore, if one accepts that the device achieves its purpose, the problem reduces to explaining how this behavior could be achieved. This simplifies the reasoning considerably since it does not have to rule out unintended behaviors. In explaining a natural system any such ambiguity has to be resolved by the introduction of new distinctions and laws. Teleology substitutes for having to analyze the system below a certain level of detail.

This discussion has left open many questions about the precise definition of teleology because the word *teleology* captures a cluster of different ideas. This chapter will investigate a number of different types of teleology and how these different types are used in recognizing circuits.

7.3 Explanation, Proof and Rationalization

The central notion in analyzing natural systems is that of proof (i.e. unique prediction). The distinctions and laws of the particular domain constitute a calculus which is utilized to prove

that the observed behavior occurs. In dealing with designed artifacts the central notion is that of rationalization: an argument in some calculus which indicates how the behavior could occur. A proof guarantees a particular behavior and rules out all others while a rationalization does neither.

In order for a particular calculus to be useful for rationalization it must be *complete, limiting* and *articulate*. A calculus is complete with respect to a domain if it is capable of expressing an argument for a wide range of behaviors that occur in the domain. A calculus is useful only if it eliminates most of the other possible behaviors, limiting the remaining possible behaviors to a small number. In order to reason about the possibilities, the calculus must articulate the source of the ambiguities. The situation-action rules that NEWTON (see chapter 3) uses are an example of a calculus that obeys these principles. Its rules account for all possible roller coaster problems, they restrict the ambiguities to a small set, and they articulate the type of the ambiguities so that quantitative reasoning can deal with it. Causal analysis fulfills a similar role in electronics, being capable of analyzing simple dc amplifiers, simple logic gates and regulated power-supplies. It arrives at a limited number of possible interpretations and these ambiguities originate from specific assumptions made in the analysis process.

In recognition, the ultimate purpose of a device can be used to identify the correct interpretation. Teleology plays a similar role in explanations of circuit function. An *explanation* is a rationalization coupled with a teleological component which, in effect, turns the rationalization into a kind of proof. The teleological component of an explanation rules out all other possible rationalizations, thus verifying the sole remaining one. The argument given for the mechanism of the Schmitt trigger meets these criteria. It contains the phrase "emitter-follower" and explicitly suggests positive feedback. Although the calculus that underlies the argument produces multiple interpretations, this teleological commentary is sufficient to identify the unique interpretation. There is only one interpretation of the Schmitt trigger which contains an emitter-follower or positive feedback (figure 1).

Although the text of the Schmitt trigger explanation contains few words which denote teleology, these words denote profound concepts which have extensive impact on the argument. The neophyte student tends to overlook these words leading him to perceive the explanation as a rationalization with the common result that he is not able to reproduce the explanation of how the circuit works. To the experienced engineer these teleological concepts immediately transform

SCHMITT-2 : ENVIRONMENT-26 = <[-E1 Q2] [Q2 VB] [-B2 RB2] [RB2 V1] [-C1 Q1] [Q1 VB] (Q2 ON) (Q1 ON); [RC1 V] [RB1 V] [RC2 I] [RE V]>

Loop:1/1; Sign:+; Comparison:LOOP; Sampling:LOOP

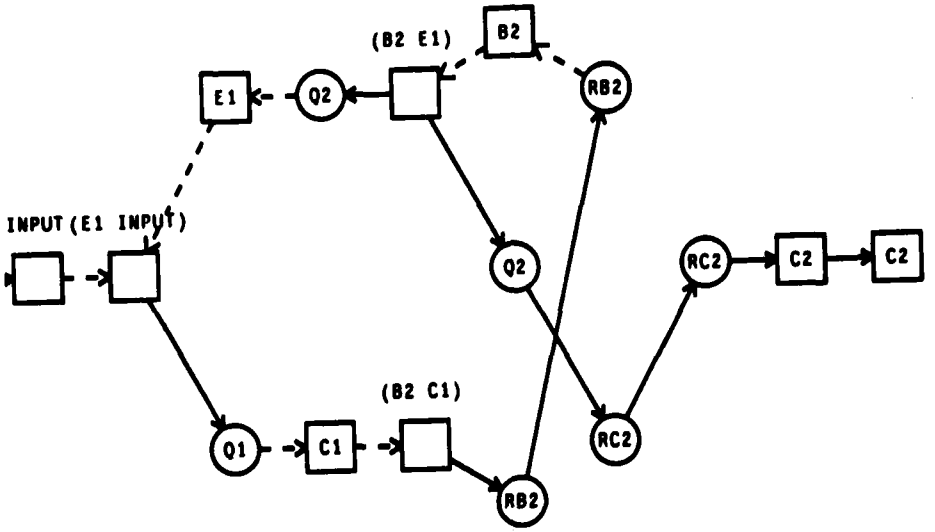


Figure 1 : Mechanism Graph of Schmitt Trigger

the rationalization into an explanation.

The explanation still does not guarantee that the circuit behaves in the stated way. An explanation is a technique by which a mechanism can be communicated and in which the author states his awareness of the ambiguity in the calculus component, resolving this ambiguity by also stating how the circuit should behave. One way to guarantee this argument is to refine the calculus to the point where the rationalization itself becomes a proof. The explanation provides the framework of a two-step type of proof. The first step is to guarantee that the rationalization indeed holds in the given interpretation. The second step is to guarantee that the mechanism meets the given teleological description which selected the unique interpretation. The calculus-teleology distinction introduces a new type of proof which first rules out all alternate hypothetical worlds and then guarantees that the rationalization holds in the given world. This topic, however, is not the subject of this research. This chapter has the more modest goal of identifying and formalizing the teleological concepts that people use in understanding circuits.

The distinction between natural systems and designed artifacts not withstanding, people often utilize teleological arguments in dealing with natural systems. This is particularly true when they try to reconstruct an argument they once knew. The final behavior may be known, and this information can be used to reconstruct a partially remembered argument. However, the final

arbiter is the distinctions and laws.

7.4 Design

Recognition can be viewed as the inverse of design. The design process produces a specific implementation which meets given purposes, while the recognition process discovers and verifies these purposes from the implementation. For this reason it is useful to examine the design process in more detail.

Design would be impossible without a language to express the purposes the artifact is to achieve. Most engineering fields have developed an extensive language to describe intent. In electrical engineering this language consists of an informal collection of types (amplifier, oscillators, power supplies, etc.) and a more formal method for stating their specifications (input impedance, transconductance, short-circuit current, etc.) These specifications refer to the parameters of components in an idealized model of the desired type of device. The problem of design is that no components exist which exactly meet these idealizations. The final device must utilize collections of existing components which individually cannot achieve the purposes, but as a composite can. The design process chooses components which meet some of the purposes and then augments these components with others to accent their desirable characteristics and suppress their undesirable characteristics.

In order to make the design process tractable, the behavior of electrical components is decomposed into two independent constituents, the quiescent and the incremental. All active electrical components are nonlinear, but often can be accurately modeled by piece-wise linear approximations. The quiescent constituent determines the appropriate linear region, while the incremental constituent deals with the perturbation behavior within the linear region. This research focuses on the incremental constituent, but many of the problems of nonideal components persist.

Some of the problems a designer must deal with are: the parameters of the devices fluctuate with manufacture, components of high enough gain do not exist, and the component does not operate at the given voltage. In ac circuits the central problem is often to achieve sufficient bandwidth. The designer also has many constraints dictated by the marketplace: the circuit must have minimal operating costs, the circuit must have minimal construction cost, and the circuit must have minimal maintenance cost. All these goals may not be simultaneously achievable and

the designer must choose the goals which are most important. These goals form the ultimate purposes for the design decisions he makes.

As the design progresses the designer chooses methods and circuit fragments which meet some of his goals. Every time he introduces a circuit fragment to the design he imposes more goals on the remainder of the design. The other unimplemented pieces must interact with this new fragment, and the fragment may have new goals it needs to have met before it can achieve its purpose. In a total design very few components can be directly related to the ultimate purpose. Usually a fragment meets a purpose of another fragment which meets a purpose of another fragment which eventually achieves a piece of the ultimate purpose.

There are many types of purposes and many types of explanations of how these purposes are achieved. The purposes of a fragment can be verified by explaining how it contributes to the functioning of the overall device. This is a *positive* explanation. The purpose of a fragment can also be explained with respect to how the overall device would fail to function if the fragment were absent. This is a *negative* explanation. The *comparative* explanation states how alternative circuit fragments do not meet the overall circuit objective as well.

Assuming the device operates in a particular way, the purposes of each of the components can be given by how they contribute to this mechanism. For example, RF in CE-FEEDBACK makes feedback possible. This is called *implementation* teleology. Devices can be described at different levels of detail, and the purposes of one level can be related to the purposes of the next level. For example, the AF amplifier of a radio makes it possible for the radio to drive the speaker. This is called *abstraction* teleology, and is the main subject of the next chapter.

The above breakdown presupposes that the purposes of the device are known and that there is a method to determine that these purposes are met. Neither is necessarily true. Some of the purposes cannot be known without consulting the designer or an expert engineer. Fortunately, this lack rarely matters; a positive explanation of the implementation teleology is usually sufficient. It is not necessary to know that feedback was utilized to minimize the effect of electrical noise in order to determine that the circuit utilizes negative feedback. This limited kind of teleological explanation is sufficient for recognition, for much of troubleshooting, and for elementary design.

7.5 Teleology and Calculus

There must be a method for determining whether a device could fulfill its purposes. A single

component may directly achieve a purpose of the overall device but the purposes are usually achieved by the synergistic action of a group of the components. This situation requires the examination of the intrinsic behaviors of the individual components to check whether the composite behavior achieves the device's purposes. A negative explanation requires that this calculus indicate the undesirable behavior that occurs when the device is absent, often necessitating a much more sophisticated calculus (e.g. consider temperature compensating diodes). For this research, I will focus on positive explanations of the implementation teleology and use causal analysis as the basis of the calculus.

Any calculus based on qualitative distinctions will be incapable of supporting some arguments, necessitating the development of a strategy to deal with this incompleteness. For example, causal analysis is incomplete in its failure to include any notion of impedance or gain. However, it does identify feedback which directly affects circuit impedance and gain. This example illustrates the role a rather coarse calculus can play in recognition, detecting features of the overall behavior which indicate the presence of more complex and interesting behavior which the calculus cannot itself explain. Causal analysis can identify the feedback which leads to changed impedances, but it cannot explain why the feedback affects the impedances. A qualitative calculus has a *dynamic* aspect, causal analysis, which identifies features in the overall behavior, and a *static* aspect, a library of declarative information, which indicates the properties the device must have as a consequence of having those features.

People regularly deal with devices whose mechanism they do not completely comprehend, or devices whose alleged behavior they might be able to verify but which they choose not to. Sometimes this is the result of *compiling* results of previous experience with the device and sometimes the internal functioning of the the device is just not understood. This type of behavior is necessary if circuits are to be understood or designed, for otherwise all reasoning would have to take place at the most detailed level. The field must have a way of summarizing its accumulated knowledge. For the sake of simplicity and efficiency the circuit should be reasoned about with the coarsest strategy that still provides a usable answer. Electrical engineering has a well-developed vocabulary of static descriptions that includes such features as feedback, coupling and bypass. These words are often used telcologically, but such uses are misleading as is illustrated by the word, feedback. Feedback is an aspect of the behavior which is known to give stable gain and one of many ways to achieve stable gain in a circuit.

The static calculus detects features in the causal mechanism and utilizes these to determine the properties of the overall device. A taxonomy of such features reveals many for which electrical engineers do not have a name. The features refer to physically realizable mechanisms which for various reasons never occur in well-designed circuits. Although a transistor appears useful as the comparator of two voltages, it is rarely used as such (except as part of a feedback system) because the result of this comparison depends critically on beta. Likewise, a transistor appears useful as an attenuator where the emitter is the input terminal and the base is the output terminal, but it is never used in this way because this behavior can be achieved more accurately with resistors. Interpretations in which some transistor solely functions in one of these ways is probably faulty. However, it is perfectly reasonable for a transistor to function both as an amplifier and in one of these two ways.

The language used to describe the mechanism and the causal analysis used to discover the mechanism, although nonteleological, have a distinct teleological bias. This might be a fatal objection to the theory developed so far since it stems from the analysis of a limited class of circuits. However, the calculus captures the teleology that is shared among all practical circuits: the theory must only be careful to exclude that teleology that could be idiosyncratic to a particular practical circuit type.

7.6 A Taxonomy of Implementation Purposes

There are only three configurations in which the transistor is capable of providing useful amplification. Each configuration has a unique combination of input impedance, output impedance, voltage gain, and current gain. The most familiar transistor configuration is the common-emitter (CE) in which the input is applied to the base and the output is taken off of the collector. The emitter provides the common terminal between input and output.

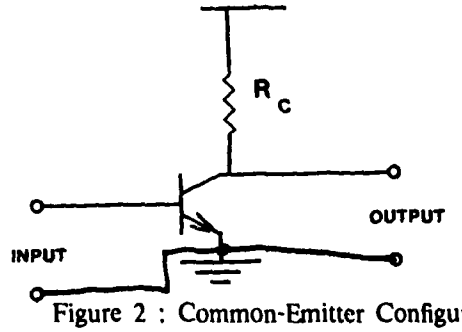


Figure 2 : Common-Emitter Configuration

This configuration has moderate input impedance and output impedance and exhibits voltage and current gain.

In the common-base (CB) configuration the input is applied to the emitter and the output is taken off of the collector. The base provides the common terminal.

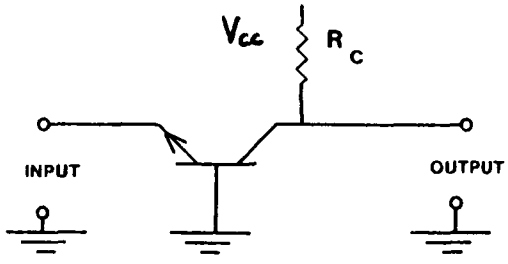


Figure 3 : Common-Base Configuration

Since the input is applied to the emitter instead of to the base the configuration has low input impedance and high output impedance. It has unity current gain but significant voltage gain.

In the common-collector configuration the input is applied to the base and the output is taken off of the emitter.

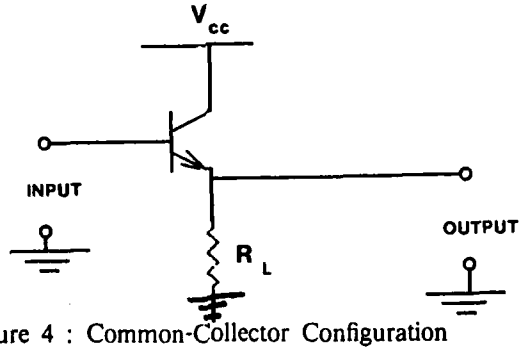


Figure 4 : Common-Collector Configuration

This configuration has low output impedance and high input impedance. It has unity voltage gain, but has significant current gain. This configuration is also called the emitter-follower configuration.

In order to avoid dealing with numerical quantities or inequalities the qualitative values UNITY, LOW, MODERATE and HIGH are used. Two qualitative values conflict when they are not neighbors (e.g. LOW conflicts with HIGH, but not with MODERATE). This algebra is very simple but adequately illustrates the teleological issues. The library characterizes an amplifier by its input impedance, output impedance, voltage-gain, and current-gain.

The specifications for the transistor configurations are:

```
CE
voltage-gain:              MODERATE
current-gain:              MODERATE
input-impedance:           MODERATE
output-impedance:          MODERATE

CB
voltage-gain:              MODERATE
current-gain:              UNITY
input-impedance:           LOW
output-impedance:          HIGH

CC
voltage-gain:              UNITY
current-gain:              MODERATE
input-impedance:           HIGH
output-impedance:          LOW
```

In order to determine how a device is utilized the mechanism graph must be consulted. The same device may be used in many different configurations in the same interpretation since it may be fulfilling multiple roles. For example, a transistor's base current is providing a load to the circuit driving it as well as amplifying this signal for the succeeding circuitry. The *causal* configuration is determined by examining the causal graphs of the circuit's outputs. The *feedback* configuration is determined by examining the feedback loop from the sampling point to comparison point. A component can also have a role in justifying an application of a connection heuristic. Such a role is a *reflection* configuration. The same device can participate in causal, feedback and reflection configurations.

The extraction of the correct configuration is not as straightforward as it may seem. Since a transistor's output is a current, the output port of the configuration is easily determined. The input port presents the difficulty. The base-emitter voltage is the transistor's only causal input. A voltage does not distinguish between its two nodes making it difficult to distinguish between the common-base and common-emitter configurations. In order to determine the input port the derivation of this voltage must be examined. If this voltage is with respect to ground, the non-ground node is the input port. If the voltage is the result of a KVL-heuristic, the node of the

KVL-heuristic is the input port. The input voltage can also be the result of a KVL deduction. If one of the two voltages of the KVL deduction is zero, the input port is the non-common node of the other voltage. If these tests fail, the transistor must be summing two independent voltages. The base-emitter voltage deduced by a KVL-heuristic is sometimes utilized by other components in addition to the transistor. In this configuration the transistor's base-emitter voltage is also considered an output.

The different input-output possibilities and their configurations are summarized in the following table:

Output:	Emitter current	Base current	Collector current	Base-emitter voltage
Input:				
Emitter voltage	?	?	CB	?
Base voltage	CC	?	CE	?
Base-emitter voltage	?	?	?	*

Table 1 : Basic Transistor Configurations

There are eight entries in the table which represent plausible behaviors for which electrical engineering has no name (the voltage-voltage configuration is an impossibility). Some of these configurations occur in practice whereas others function so poorly that they are never utilized.

In order to complete this taxonomy some new nomenclature needs to be introduced. The configuration in which a transistor compares two independently changing voltages producing a collector current is called SUM. In the remaining configurations the transistor is being utilized as a kind of a diode. A two-terminal device can be used to couple two circuit fragments (COUPLE), to present a load to another fragment (LOAD), or to sense a voltage or current (SENSE). See table 2. The SUM configuration is never used because the result of the summing is dependent on beta and the circuit presents different impedances to the voltages it is comparing. Therefore every interpretation in which a transistor is operating in the SUM configuration is highly suspect. Since transistors are intended to amplify, any interpretation in which the transistor only appears in two-terminal configurations (COUPLE-Q, SENSE-Q or LOAD-Q) is also considered suspect. (QUAL translates transistors topologically connected as diodes to diodes.)

Output:	Emitter	Base	Collector	Voltage
Input:				
Emitter	LOAD-Q	COUPLE-Q	SENSE-Q	
Base	CC	LOAD-Q	CE	SENSE-Q
Voltage	SENSE-Q	SENSE-Q	SUM	*

Table 2 : Transistor Configurations

The engineering nomenclature for the configurations of two-terminal devices is not as precise. The three configuration categories are COUPLE, LOAD, and SENSE. Since the resistor can transform voltages into currents and currents into voltages, these categories are further broken down into the particular type of sensing, coupling or loading. The voltage input terminal is determined in the same way as the input terminal of a transistor. If one of the terminals is fixed, the other terminal is considered to be voltage output.

Output Current:	#1	#2
Input:		
Current at #1:	*	WIRE
Current at #2:	WIRE	*
Voltage at #1:	V-LOAD	V-TO-I-COUPLE
Voltage at #2:	V-TO-I-COUPLE	V-LOAD
Voltage between #1 and #2:	V-SENSE	V-SENSE

Output Voltage:	#1	#2	#1 and #2
Input:			
Current at #1:	I-LOAD	I-TO-V-COUPLE	I-SENSE
Current at #2:	I-TO-V-COUPLE	I-LOAD	I-SENSE
Voltage at #1:	*	*	DIVIDER
Voltage at #2:	*	*	DIVIDER

Table 3 : Resistor Configurations

The LOAD configurations utilize the same terminal for the input and output signals. The other terminal is held fixed.

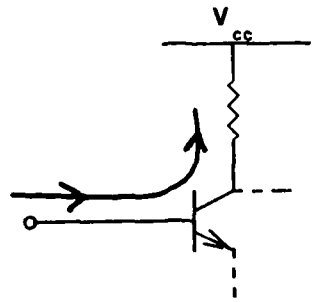


Figure 5 : Example of the I-LOAD Configuration

The effect of the transistor's changing collector current is first detected at the lower resistor terminal. This current produces a voltage drop across the resistor. Since the other end of the resistor is held fixed, the voltage produced is evident at the lower terminal.

A voltage at one terminal of a resistor can cause a current out of that same terminal. In this V-LOAD configuration the output terminal is determined by the mechanism graph.

The COUPLE configurations utilize opposite terminals for input and output signals.

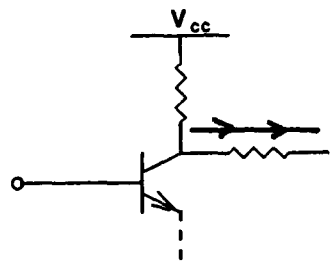


Figure 6 : Example of the V-TO-I-COUPLE Configuration

In this example, an application of a KCL-heuristic at the collector produces a voltage which a subsequent KVL-heuristic applies to the resistor (causing reflections). Thus the left-hand terminal can be viewed as the input. The right-hand terminal must be the output since the other terminal current is only utilized to justify the KCL-heuristic. Analogously, I-TO-V-COUPLE is also possible.

The remaining resistor configurations are less common. In the WIRE configuration, the voltage-current transformation capacity of the resistor is not used and the component is only used in a KCL deduction.

In the SENSE and DIVIDER configurations unique input-output terminals cannot be assigned. In the V-SENSE configuration a voltage across the resistor produces a current through it. More commonly, the current through a resistor is sensed by the voltage across it. This is the I-SENSE

configuration. An application of a KVL-heuristic can result in a voltage across the resistor which is directly utilized by other components. This is the DIVIDER configuration. There are a few more theoretically possible configurations, but they never arise.

The resistor configurations are another example where the analysis depends on a notion of a common fixed ground. If there were no common ground, it would be more difficult to detect LOAD configurations, and these would be analyzed as SENSE configurations.

Using this taxonomy QUAL identifies the configurations (assuming the correct interpretation) for the Schmitt trigger as follows:

```
Q1 used as CE in CAUSAL configuration.
RC1 used as I-LOAD in REFLECTION configuration.
RB2 used as I-LOAD in REFLECTION configuration.
RB2 used as V-TO-I-COUPLE in CAUSAL configuration.
RB1 used as I-LOAD in REFLECTION configuration.
Q2 used as LOAD-Q in REFLECTION configuration.
Q2 used as CC in FEEDBACK configuration.
Q2 used as CE in CAUSAL configuration.
RE used as I-LOAD in REFLECTION configuration.
RC2 used as I-LOAD in CAUSAL configuration.
```

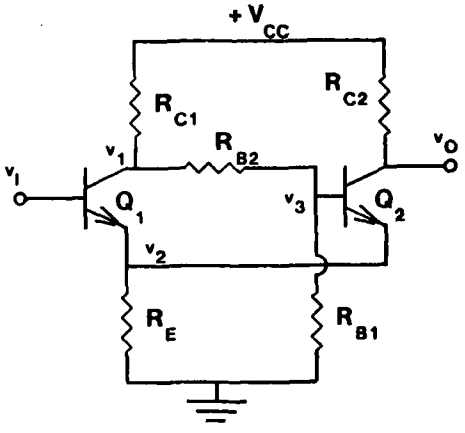


Figure 8 : Schmitt Trigger

A single component can appear in different configurations. For example, Q2 appears as a load

to the coupling circuit (LOAD-Q), as a common-emitter output stage (CE), and as a common-collector stage on the feedback path (CC).

The correct reflection configuration is determined by applying the causal configuration patterns and inverting the result. The KCL-heuristic at C1 applies a voltage to RC1 and the resulting current substantiates the heuristic. Thus RC1 is used in a V-LOAD configuration. The connection heuristics are an artifact of the analysis process and the canonical configuration is derived from what it would be if the heuristics were unnecessary. Thus RC1 appears in the I-LOAD configuration.

7.7 Amplifier Recognition

The input-output behavior of an amplifier can be specified by its input impedance, output impedance, type(s) of gain, and stability of gain. This simple specification language provides a mechanism-independent way of describing behaviors which can be used for circuit fragments as well as entire circuits. The basic circuit fragment consists of a single component in a particular configuration. QUAL utilizes a library to determine the specifications of these component configurations. The specifications of the individual configurations are combined to compute the specifications of the entire circuit. For example, a stage with voltage gain but unity current gain followed by a stage with current gain but unity voltage gain results in a composite circuit which has both voltage gain and current gain.

In order to determine the specifications of a circuit, QUAL constructs a new representation of the circuit's behavior. Starting with the mechanism graph which describes the events that take place in an operating circuit, collections of events representing the functioning of particular configurations are combined into fragments. Every vertex and edge in the mechanism graph is included in some fragment such that the resulting representation can be reasoned about independently of the original mechanism graph. A *fragment* represents a connected subgraph of the mechanism graph and is described by input and output ports which connect to other fragments. Each port is connected by a link to a unique port in another fragment. Since a fragment represents a piece of the mechanism and not a piece of topology, the same component can participate in multiple fragments.

The configurations for the components account for most of the mechanism graph. Special INPUT and OUTPUT fragments are necessary to account for circuit inputs and outputs. Four primitive fragments originate from feedback and feedforward: SPLIT, JOIN, COMPARISON

and SAMPLING. The final primitive fragment, GLUE, is used to account for paths within the mechanism graph which are not accounted for by any of the component fragments. The result of this process is a description of the mechanism by which the circuit achieves its behavior that is more amenable to hierarchical analysis.

The primitive fragments for CE-FEEDBACK are:

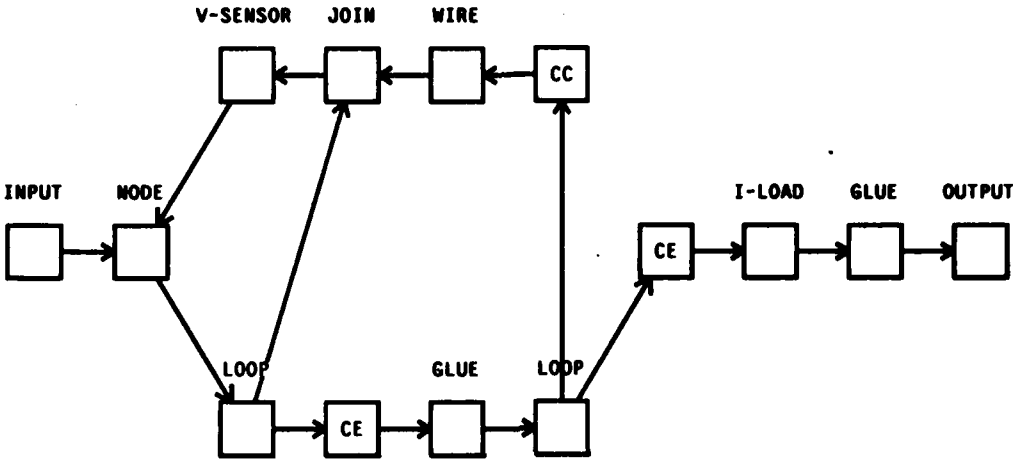


Figure 9 : Primitive Fragments of CE-FEEDBACK

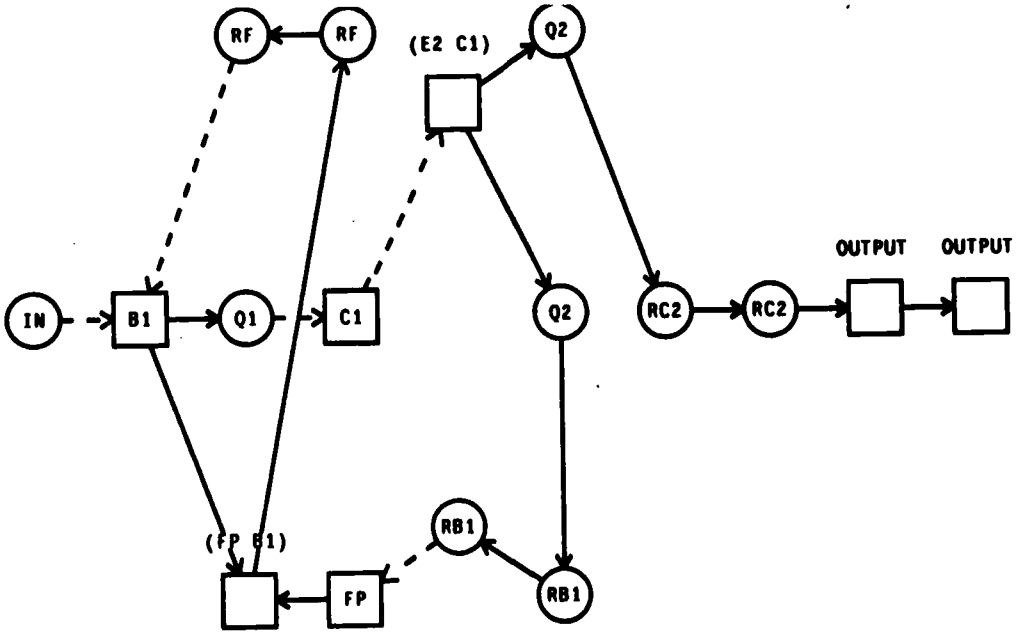


Figure 10 : Mechanism Graph for CE-FEEDBACK

Recognition proceeds by applying fragment substitution rules until only one fragment remains. Through this substitution process a hierarchical description of the circuit's mechanism is constructed. Six rules are sufficient to recognize simple amplifiers:

1. Ignore fragments of passive components.
2. Combine cascaded stages.
3. Remove local feedback.
4. Modify sampling points.
5. Collapse feedback loops.
6. Detect top-level circuit configuration.

The first rule removes all device fragments which do not originate from active devices such as transistors, and all GLUE fragments. This rules assumes that these fragments do not contribute significantly to circuit behavior. The last rule detects the terminating situation where all but one of the fragments are INPUTs or OUTPUTs.

Two connected active device configurations (called stages) can be combined into a single fragment. The specifications of the composite stage are computed from the specifications of the constituent stages. The gain of the composite is the product of the constituent gains. The stability of a particular gain is preserved only if both stages stabilize the same gain type. The input impedance of the composite fragment is the input impedance of the input stage and the output impedance of the composite fragment is the output impedance of the output stage. A check is also made to verify that the input impedance of the output stage matches the output impedance of the input stage. (In some circuits this mismatch is desirable — the designer is trading off stability against power gain.)

Consider the following simple video amplifier:

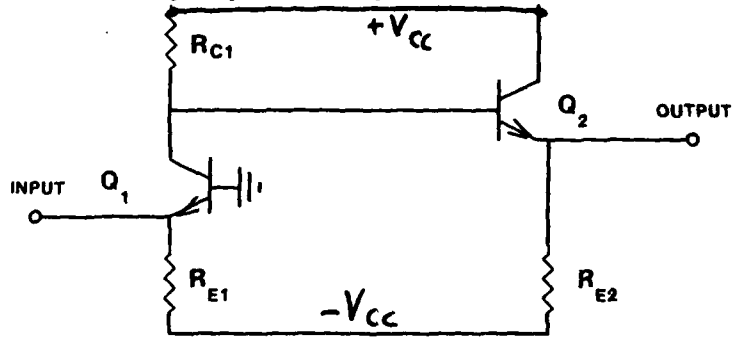
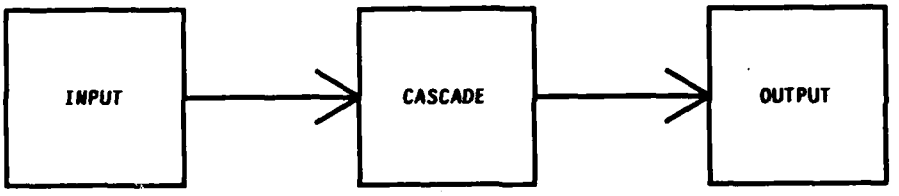


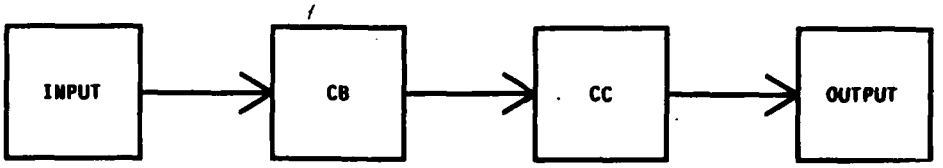
Figure 11 : VIDEO

The VIDEO amplifier can be analyzed by one application of each of the above rules:

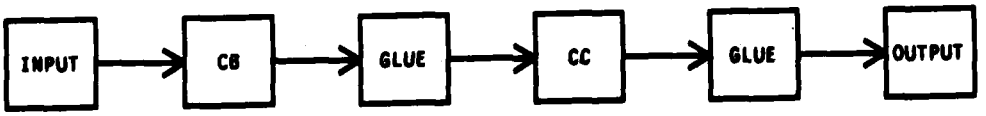


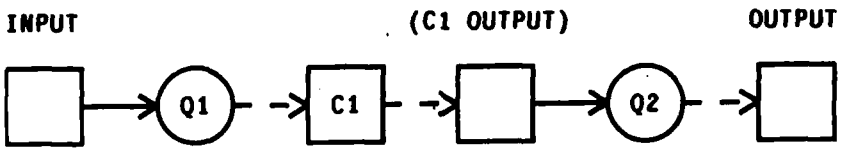

```
INPUT-IMPEDANCE : LOW
OUTPUT-IMPEDANCE : LOW
CURRENT-GAIN : MODERATE
VOLTAGE-GAIN : MODERATE
```

```
STABLE : VOLTAGE-GAIN
VOLTAGE-GAIN : UNITY
CURRENT-GAIN : MODERATE
INPUT-IMPEDANCE : HIGH
OUTPUT-IMPEDANCE : HIGH
```




```
STABLE : CURRENT-GAIN
VOLTAGE-GAIN : MODERATE
CURRENT-GAIN : UNITY
INPUT-IMPEDANCE : LOW
OUTPUT-IMPEDANCE : HIGH
```






```
VIDEO-2 : ENVIRONMENT-7 = <[+OUTPUT Q2] [Q2 VB] [+C1 Q1] (Q2 ON) (Q1 ON);
[RE2V] [RC1 V] [Q1 V] [RE1 V]>
```

Figure 12 : Parse of VIDEO

The remaining three fragment substitution rules apply to feedback. Feedback presents an

immediate problem for fragment construction since the sampling and comparison vertices can occur within configurations. For example, if a SAMPLING (or SPLIT) occurs between the emitter and collector there are events before and after the SAMPLING that belong to the two transistor configurations. Therefore, SPLITs, SAMPLINGs, and COMPARISONs, all "move" to their nearest antecedent vertex which occurs at a fragment boundary. In a few cases this modification is insufficient for recognizing SAMPLINGs and a special fragment substitution rule is included for these.

The analysis of CE-FEEDBACK requires all three feedback rules.

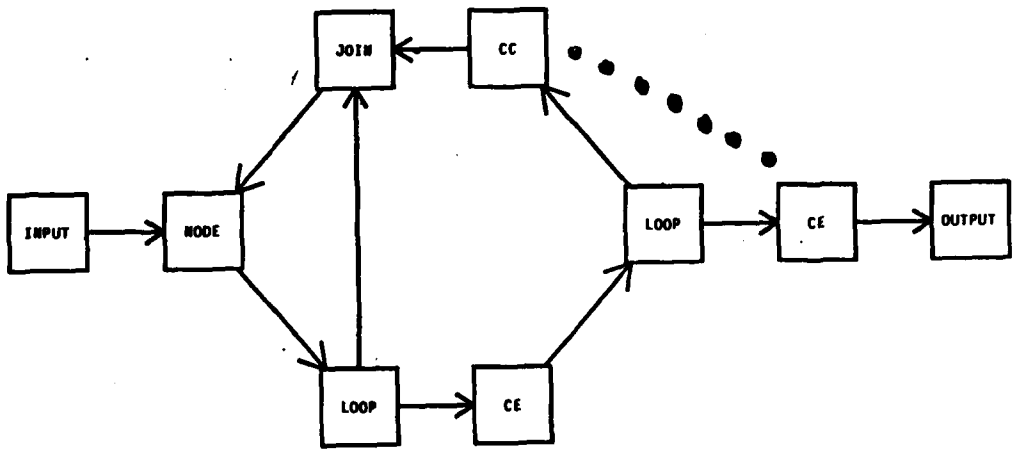


Figure 13 : Reduced Fragments for CE-FEEDBACK

Since the overall feedback is through RF the voltage at the base of the input transistor also contributes to the current through RF. Since this contribution can be ignored, the local feedback loop it produces must be removed.

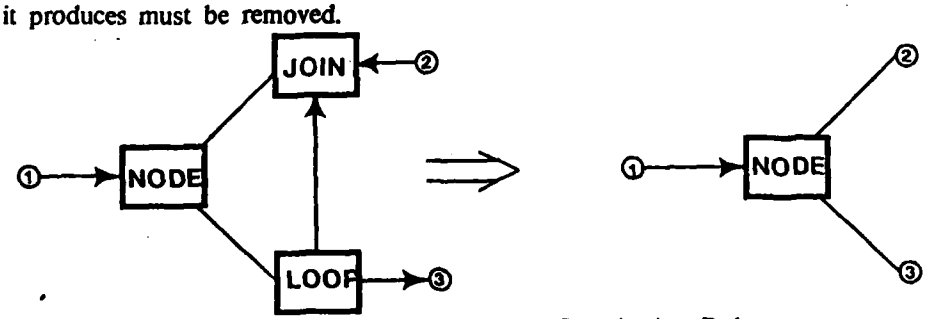


Figure 14 : Local Feedback Substitution Rule

In the fragment graph, Q2's CE configuration is outside of the feedback loop:

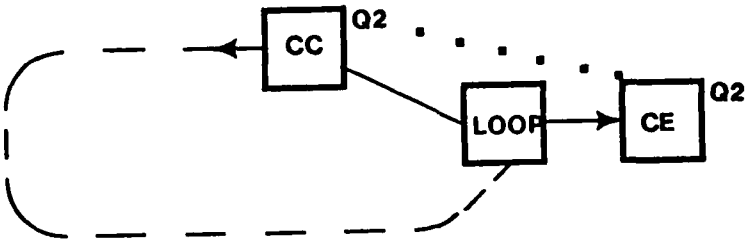


Figure 15 : Q2 configuration

The feedback action also affects the CE stage. Placing the CE stage outside of the feedback loop may also cause the cascade rule to combine it with another stage. A better representation is to include the stage within the feedback loop.

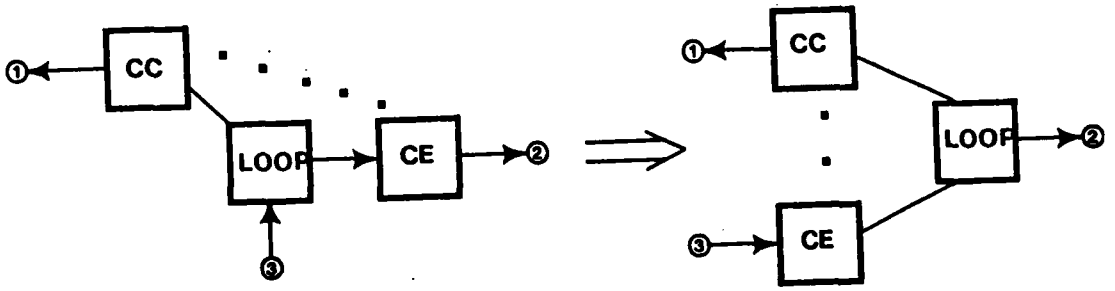
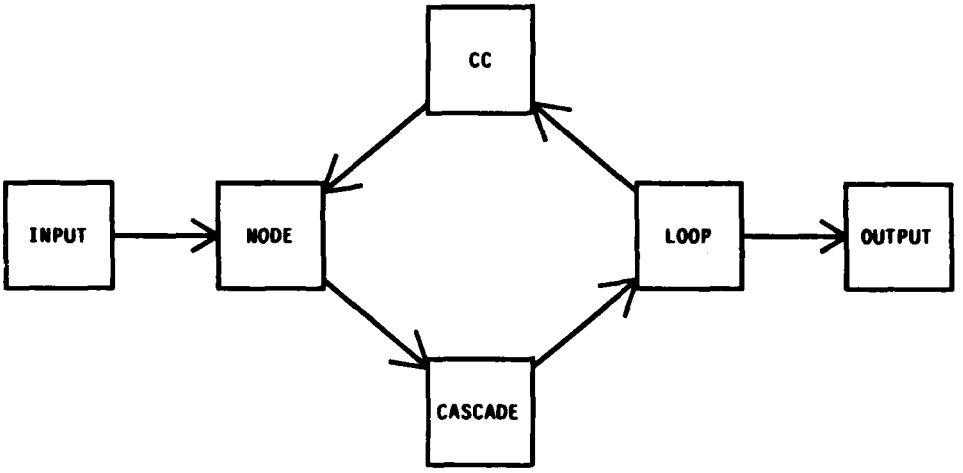


Figure 16 : Sampling Rewrite Rule

This rule makes it easier to check impedances and determine the effects of feedback, but it suggests an incorrect gain around the feedback loop. There is no way to represent circuit mechanism in this graphical notation which makes it possible both to compare impedances by considering adjacent stages and to compute gains by tracing stages. Since QUAL does not look at feedback loop gains, this representation is adequate.

After applying all the rules the fragments of CE-FEEDBACK are:




```
INPUT-IMPEDANCE : MODERATE
OUTPUT-IMPEDANCE : MODERATE
CURRENT-GAIN : HIGH
VOLTAGE-GAIN : HIGH
```

Figure 17 : Fragments of CE-FEEDBACK

All feedback loops are of the form:

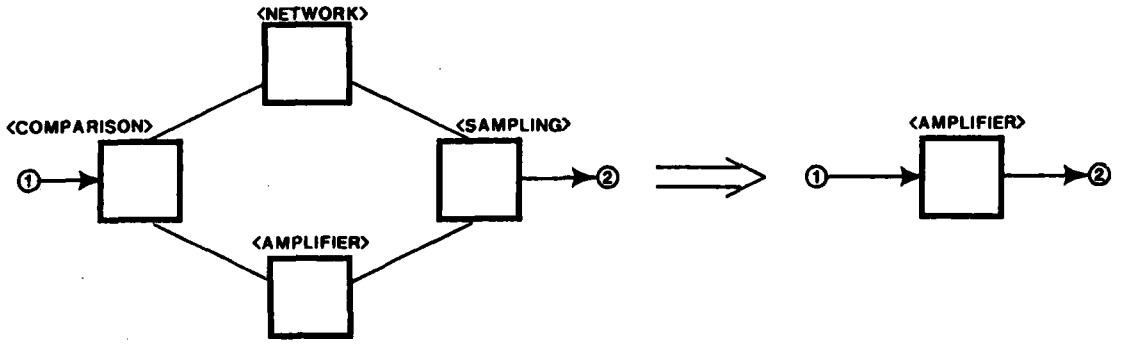
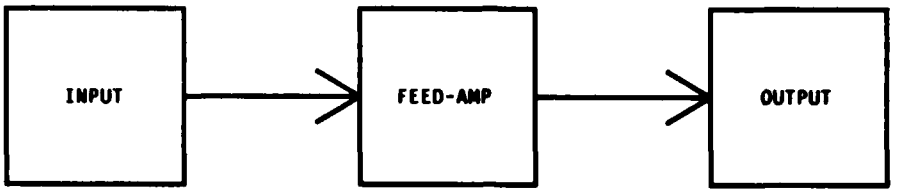


Figure 18 : General Feedback Loop

The effect of feedback on amplifier gain and impedance is presented in Table 6.1. In the case of CE-FEEDBACK the feedback altered specifications are:




```
INPUT-IMPEDANCE : MODERATE
OUTPUT-IMPEDANCE : MODERATE
CURRENT-GAIN : HIGH
VOLTAGE-GAIN : HIGH
```

```
STABLE : CURRENT-GAIN
INPUT-IMPEDANCE : LOW
OUTPUT-IMPEDANCE : HIGH
CURRENT-GAIN : HIGH
VOLTAGE-GAIN : HIGH
```

Figure 19 : Fragments of CE-FEEDBACK

This final fragment graph is detected by the termination rule.

The algebra used for impedance calculations uses the values LOW, MODERATE and HIGH. For impedance computations only a LOW-HIGH match is considered grounds for complaint. The feedback action can change the impedance one step in this metric. This algebra is simple but it suffices for simple amplifiers.

7.8 Disambiguating Interpretations

It is often not necessary to know the ultimate purpose of a circuit in order to disambiguate its correct interpretation Just the knowledge that the circuit has some purpose can be enough to identify its correct interpretation, the implementation purpose forming the basis by which to do the disambiguation. Since every component has a purpose, the correct interpretation should assign purposes to as many components as possible. Therefore, if the purposes of one interpretation are a subset of some other, the first interpretation should be disregarded. An interpretation with

more implausible configurations than other interpretations should also be disregarded. These two rules almost always select a unique interpretation.

It is surprising that the correct interpretation of a circuit can be determined with just these two rules. The reason for the success of the strategy is twofold. First, pragmatic considerations such as cost and complexity dictate that only a small subset of the space of possible circuits is ever designed. These considerations (which we know) are part of the circuit's ultimate purpose. Second, the taxonomy of implementations is a formalization of the compiled experience of electrical engineers. Assuming that this taxonomy is complete, unrecognized configurations can be *a priori* presumed to be useless. If the circuit contained numbers of irrelevant components (such as a TV inside of a telephone circuit) the first rule fails. The second rule fails if a circuit utilizes an old component in a new way.

Three simple, nearly independent rules for determining the correct interpretation are:

1. Choose the interpretation with maximum "purpose."
2. Choose the interpretation with fewest implausible configurations.
3. Choose the interpretation which exhibits feedback.

The last method fails for more general circuits. However, a combination of the the first two is successful for a fairly wide range of circuits.

A component can be used in a feedback configuration, causal configuration, or reflection configuration, or be unused. Feedback and causal configurations directly affect circuit specifications. Reflection configurations contribute by supporting connection heuristics. Unused components play no role. This ordering on configurations imposes a partial order on interpretations. In the cases where this order has a unique maximum, this maximum is the correct interpretation.

CE-FEEDBACK has four interpretations: correct (figure 6.9), feedbackless (figure 6.32), feedforward (figure 6.28) and unity-gain (figure 6.30). Table 4 lists the implementation purposes of all the components. "(C)", "(F)" and "(R)" indicate causal, feedback and reflection configurations. The two maxima are the correct and feedforward interpretations. Since the feedforward interpretation utilizes Q2 in an implausible configuration, it is ruled out. This choice could also be made by choosing feedback over feedforward interpretations. For CE-FEEDBACK, choosing feedback interpretations over all other interpretations also gives the correct interpretation.

Configuration:	Correct	Feedbackless	Feedforward	Unity gain
Component:				
Q1	CE(C)	CE(C)	CE(C)	LOAD-Q(R)
	LOAD-Q(R)	LOAD-Q(R)	LOAD-Q(R)	
Q2	CE(C)	CE(C)	SUM(C)	CB(C)
	CC(F)	LOAD-Q(R)	SENSE-Q(R)	
	LOAD-Q(R)			
RC1	I-LOAD(R)	I-LOAD(R)	I-LOAD(R)	I-LOAD(R)
RC2	I-LOAD(C)	I-LOAD(C)	I-LOAD(C)	I-LOAD(C)
RB2	I-LOAD(R)	I-LOAD(R)	I-LOAD(R)	I-LOAD(R)
RB1	WIRE(F)	?	V-TO-I-COUPLING(C)	V-TO-I-COUPLING(C)
			I-LOAD(R)	I-LOAD(R)
RF	V-SENSOR(F)	I-LOAD(R)	V-TO-I-COUPLING(C)	V-TO-I-COUPLING(C)
	I-LOAD(R)		I-LOAD(R)	I-LOAD(R)

Table 4 : Interpretation Configurations for CE-FEEDBACK

The following amplifier, DEVDET, employs a common-collector stage in the feedback network. It has five interpretations. Four of these are analogous to those of CE-FEEDBACK and the fifth originates from an ambiguity around node B2. The interpretations are even easier to disambiguate than CE-FEEDBACK's since a signal flowing in the reverse direction through the feedback network has Q2 operating in the implausible COUPLE-Q configuration.

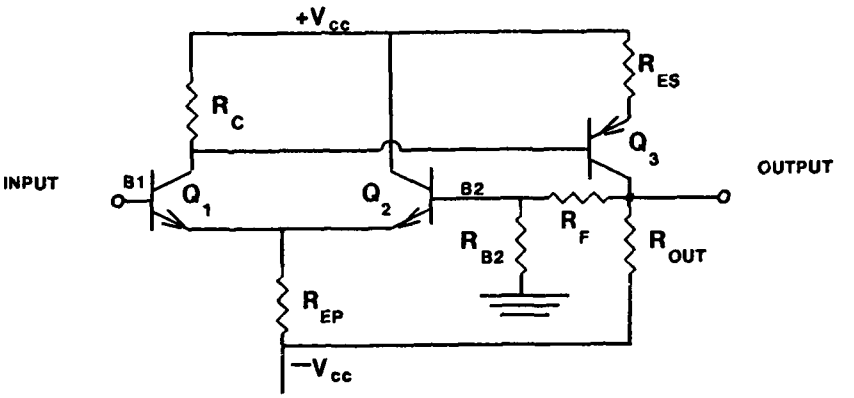


Figure 20 : DEVDET

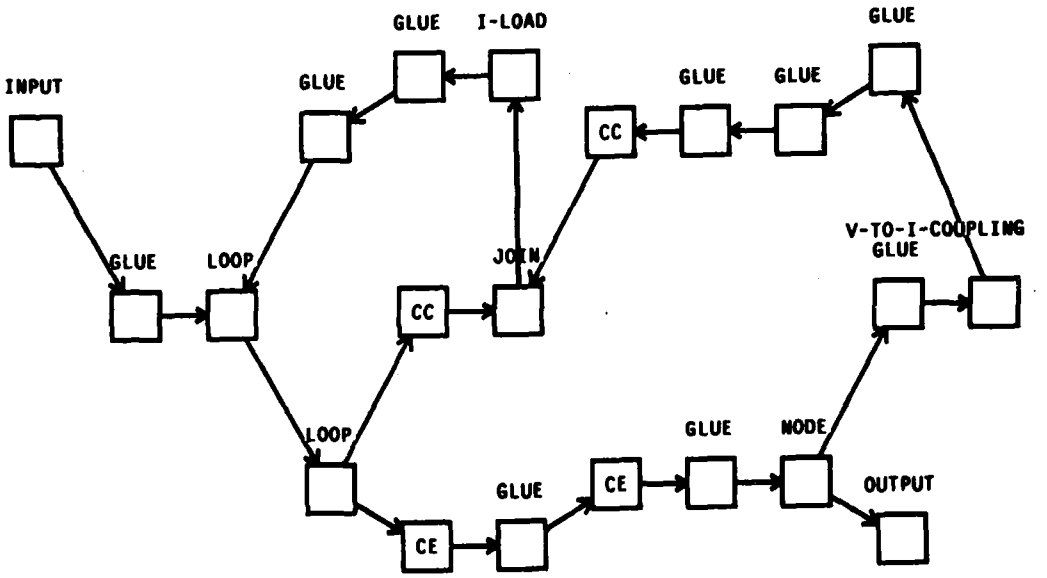


Figure 21 : Stages of DEVDET

7.9 Positive and Negative Purpose Descriptions

The hierarchical description generated by the fragment substitution rules in determining the specifications of the amplifier provides a positive explanation as to how each particular component contributes to the amplifier's overall behavior. Since the amplifier recognizer focuses on active devices, the purpose of passive devices must usually be determined by considering their associated active device.

The following is QUAL's explanation of the positive purpose of each of the components in CE-FEEDBACK. Reflection configurations are left out unless the component is functioning only in a reflection configuration (i.e. LOAD-Q for Q2 is left out, while I-LOAD for RC1 is included).

```
Q1 is functioning in CE configuration.
Which is STAGE1 of CASCADE
Which is BASIC-AMPLIFIER of FEEDBACK
Which is STAGE of TOP-LEVEL

Q2 is functioning in LOOP configuration.
Which is SAMPLING of FEEDBACK
Which is STAGE of TOP-LEVEL
And,
Q2 is functioning in CE configuration.
Which is STAGE2 of CASCADE
Which is BASIC-AMPLIFIER of FEEDBACK
Which is STAGE of TOP-LEVEL
And,
Q2 is functioning in CC configuration.
Which is FEEDBACK-COUPLING of FEEDBACK-NETWORK
Which is FEEDBACK-NETWORK of FEEDBACK
Which is STAGE of TOP-LEVEL
```

```
RC1 is functioning in I-LOAD configuration.
For Q1 functioning in CE configuration.
Which is STAGE1 of CASCADE
Which is BASIC-AMPLIFIER of FEEDBACK
Which is STAGE of TOP-LEVEL

RC2 is functioning in I-LOAD configuration.
Which is COUPLING of OUTPUT-NETWORK
Which is OUTPUT-NETWORK of TOP-LEVEL

RB1 is functioning in WIRE configuration.
Which is FEEDBACK-COUPLING of FEEDBACK-NETWORK
Which is FEEDBACK-NETWORK of FEEDBACK
Which is STAGE of TOP-LEVEL

RB2 is functioning in I-LOAD configuration.
For Q2 functioning in CC configuration.
Which is FEEDBACK-COUPLING of FEEDBACK-NETWORK
Which is FEEDBACK-NETWORK of FEEDBACK
Which is STAGE of TOP-LEVEL

RF is functioning in V-SENSOR configuration.
Which is FEEDBACK-COUPLING of FEEDBACK-NETWORK
Which is FEEDBACK-NETWORK of FEEDBACK
Which is STAGE of TOP-LEVEL
```

Since the recognizer can determine the specifications of any amplifier, a negative purpose description can be constructed by removing the component and determining the specifications of this modified amplifier.

```
If Q1 is collector-opened:
CURRENT-GAIN changed from HIGH to UNITY
VOLTAGE-GAIN changed from HIGH to MODERATE
(Q1 only used in improbable configurations.)
```

```
If RB2 is shorted:
Circuit lost STABLE CURRENT-GAIN
INPUT-IMPEDANCE changed from LOW to MODERATE
OUTPUT-IMPEDANCE changed from HIGH to MODERATE
```

Since QUAL constructs a multi-level representation of the circuit's behavior, the desired and actual behaviors can be compared at many levels. A comparison at the configuration level detects that opening Q1 forces Q2 into the common-base configuration from the common-emitter configuration. Similarly, a comparison at the mechanism graph level detects that a shorted RB2 prevents the desired feedback action from happening. There is no point to extending QUAL's explanation capabilities any further without researching the nature of the explanation. The above two examples illustrate only some of the possibilities.

Topological construction, causal analysis, interpretation construction, feedback analysis, and teleological analysis all impose criteria which rule out implausible interpretations. Criteria imposed by topological construction and causal analysis prevent implausible interpretations from being constructed, and interpretation construction, feedback analysis, and teleological analysis rule out constructed interpretations. When these criteria rule out all of the interpretations, something is wrong with the circuit itself. In order to determine the negative purpose descriptions QUAL substitutes short or open circuits for the components and sees how the amplifier's specification change. Usually QUAL cannot analyze the resulting circuit since the criteria rule out all interpretations. Shorting RC2 causes output voltage to be pinned to vcc. When RC2 is open there is no way for the circuit to affect its output. Similarly for RB1. Therefore the only negative purpose description QUAL can generate for RC2 is that it must be present for the circuit to work. This is true, but not illuminating.

If possible, QUAL does not rule out the last interpretation, and instead accumulates the complaints about it. In the above example, when RC1 is shorted, QUAL notices that Q1 is not in a plausible configuration in the only remaining interpretation.

The criteria could be used to check tentative designs. A design system would present a circuit with an alleged output behavior and QUAL could determine whether those specifications were met, and what complaints it had about the circuit. The design system could then utilize these complaints to produce an improved circuit.

Although QUAL's criteria enable it to select correct interpretations, they are insufficient to recognize malfunctioning circuits. QUAL will recognize as amplifiers many circuits which cannot

possibly amplify. In most cases these circuits are so badly designed that they could have no useful purpose anyway. However, this weakens QUAL's usefulness for design. For example, QUAL detects no change in CE-FEEDBACK's specifications if RB2 is opened, RB1 is shorted or RF is shorted. None of these modified circuits can possibly do anything useful. A bias analysis would detect some of these faulty circuits, but both types of analysis are necessary. For example, RB2 open is implausible since it removes the bias supply for Q2. RB1 shorted is implausible since it moves the feedback stability point outside of the active region of the transistors.

7.10 *Applications to SYN

Certain combinations of transistors occur so frequently that it is useful to consider them as single fragments. A common-collector stage followed by a common-emitter stage forms an amplifier with high input impedance and moderate output impedance. This combination is known as a Darlington pair. The common-emitter-common-base combination is called a cascode and has very good frequency response. The common-collector-common-base connection forms a circuit widely used in operational amplifiers. The emitter-coupled pair provides differential outputs and can be direct coupled. The recognition of these combinations is not critical to QUAL since it can calculate the impedance-gain specifications of these combinations with its composition rules, but it can be of use to SYN.

Since SYN does not know how a particular component is being used, it must employ the most complex model available in its quantitative analysis. In most situations, however, more simple models are sufficient. The complete hybrid-π model for a transistor is:

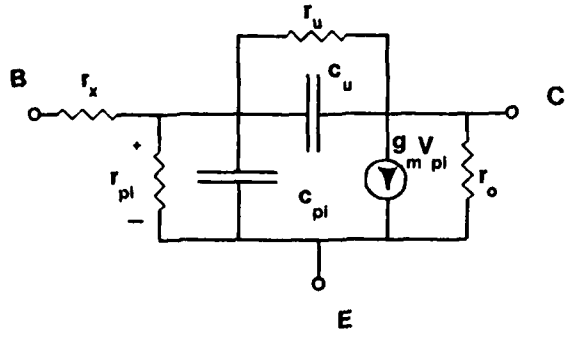


Figure 22 : Complete Hybrid-π Model

QUAL can give advice to SYN as to how a component is being used so that it can choose an appropriate simplification of the complex model.

At reasonable currents $r_x \ll r_\pi$ and can be ignored unless the circuit is driven by a low source impedance. r_μ and r_o can be ignored unless the load is of extremely high impedance. Since only the CE configuration is ever used with extremely high loads, r_μ and r_o can be ignored in the CB and CC configurations. c_μ can be ignored unless the circuit is of high impedance. When the circuit is being driven by a very low impedance c_π can be ignored. In the common-base configuration c_μ is usually not important.

Common combinations such as cascode, differential pair, and darlington have well known rules-of-thumb for making their quantitative analysis easier. QUAL can recognize these situations for SYN.

The hybrid-π model is particularly useful for analyzing the common-emitter and common-collector configurations. In the common-base configuration a different equivalent model makes the analysis easier. The T model is equivalent to the hybrid-π model, but it usually requires one less variable to solve.

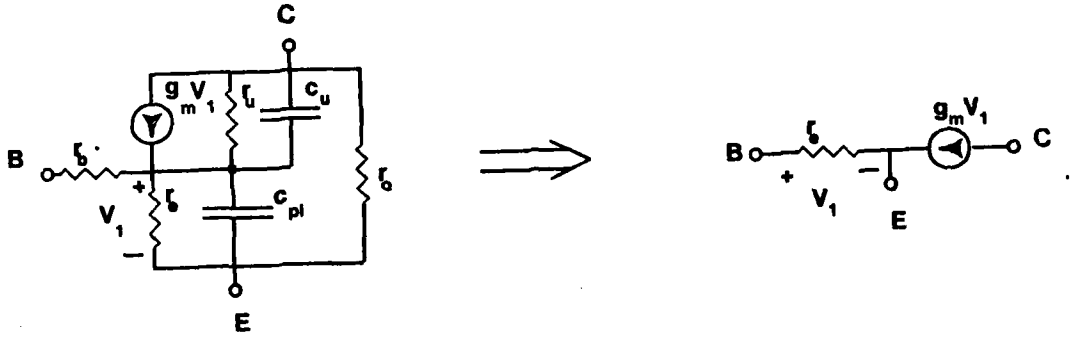


Figure 23 : T models

Chapter 8

ABSTRACTION

8.1 Bottom-Up Abstraction Teleology

The final step of the recognition process constructs a hierarchical representation of how the circuit achieves its ultimate purpose. The groundwork for this abstraction teleology as well as the abstraction rules for simple amplifiers were presented in previous chapter. The current chapter presents a larger collection of rules which are applicable to more complex amplifiers as well as to power-supplies. Although current sources and voltage sources cannot be successfully analyzed by QUAL's causal analysis, their presence can be detected and dealt with. This chapter concludes with a detailed summary of the recognition process by demonstrating QUAL's behavior on the 3rd scenario of chapter 1.

In the first two steps of the recognition process causal analysis on the circuit's topology constructs a primitive representation of the circuit's abstract mechanism. The final step performs topological manipulations on this representation of circuit mechanism in order to construct a hierarchical decomposition of the mechanism into a small collection of standard circuit types. This report focuses on the first two steps. This chapter presents a few ideas about the last step, but they have not been worked out in detail (however, QUAL did all the examples). This chapter should not be regarded as contributing new ideas to the theory, but rather as illustrating the role the representation of circuit mechanism can play in identifying the circuit. Although many researchers have studied topological matching and abstraction teleology, I have not taken advantage of that research, choosing instead the simplest and most direct way of illustrating the basic point. Hence the brevity of this chapter.

The *intrinsic* specifications of a circuit fragment are given by its voltage gain, current gain, input impedance, and output impedance. These specifications characterize the behavior of the fragment independent of any context. However, each fragment contributes to the overall functioning of the circuit, and this contribution is characterized by its *extrinsic* specifications. These specifications

describe how the intrinsic behavior of the subfragments contribute to the intrinsic behavior of the parent.

Larger fragments are constructed from smaller ones by *abstraction rules* which describe how collections of fragments are combined. These abstraction rules provide the extrinsic specifications of the fragments which they combine. The rules are divided into two general classes: those which compute the specifications of the parent from the specifications of the subfragments, and those which utilize information embedded in the names and connectivity of the subfragments. The amplifier cascade rule is an example of the former since it multiplies the gains of successive stages to find the gain of the parent. The feedback rule is an example of the latter since it takes into account the loop structure of the fragment graph. The rules presented in this chapter tend to be of the latter type.

Each fragment has a *class* and a *type* and where relevant, impedance and gain specifications. For the primitive fragments associated with components, the type is the configuration name. The ontology arrived at in the previous chapter is:

class	types	description
IO	INPUT,OUTPUT	*signals on boundary*
SAMPLING	NODE,LOOP	*feedback sampling point*
COMPARISON	NODE,LOOP	*feedback comparison point*
SPLIT	*unused*	*signal splits n ways*
JOIN	VOLTAGES,CURRENTS	*two signals combine*
STAGE	CE,CC,CB,CASCADE,FEEDBACK	*amplifying stage*
DIFF-2-1	SUM	*differential amplifier*
COUPLING	GLUE,V-LOAD,WIRE,I-LOAD, V-SENSOR,V-TO-I-COUPLING, I-TO-V-COUPLING,SENSE-Q,LOAD-Q, COUPLE-Q	*signal couplers*

Only STAGE and DIFF-2-1 fragments possess gain and impedance specifications. If a stage has multiple inputs and outputs, separate impedance and gain specifications are recorded for each.

Each rule assigns role names to each of its subfragments, and when QUAL is asked for the

purpose of a fragment or component it prints out that role name, the parent's class and type, and the name of the abstraction rule that applies:

```
Which is <role> of <type> <class> recognized by <rule>.
Which is STAGE1 of CASCADE STAGE recognized by CASCADE.
```

In many cases this form of explanation is redundant and the following format, used in chapter 7, is more succinct:

```
Which is <role> of <rule>.
Which is STAGE1 of CASCADE.
```

It is unnecessary to know the specific ultimate purpose of the circuit for the class of circuits QUAL attempts to recognize. QUAL can determine the type of the circuit through bottom-up rules; top-down analysis is never necessary. This is not true for circuits in general, and the rules that QUAL uses are carefully ordered to achieve this bottom-up behavior. However, within each general class of circuits a carefully chosen library can select the precise type within the class. QUAL's rule library is carefully chosen to work for amplifiers and power supplies, and this library is used to identify the specific type of the power supply or amplifier.

8.2 *Voltage and Current Sources

Since the central purpose of a voltage or current source is to have no incremental output, they are difficult to analyze using IQ analysis. Sources can be grouped into three general categories: *basic*, *modified-basic* and *feedback*. A *basic* source utilizes a single component whose intrinsic specifications directly match the intrinsic specifications of a source (e.g. a battery or zener diode). A *modified-basic* source consists of a basic source cascaded with a nonfeedback amplifier. The purpose of the amplifier is to extend the operating region of the basic source. A *feedback* source utilizes feedback to stabilize the output voltage or current of a another source. Causal analysis can detect the presence of sources, and can analyze basic and feedback sources which directly contribute to a stage on the mechanism graph, but it cannot analyze modified-basic sources. The area of a feedback source which contains a modified-basic source is not analyzable with IQ

analysis. This section describes how basic and modified basic sources can be detected and how this information is included in the teleology of the circuit. Feedback sources will be considered in the section on power-supplies.

Current sources are used to provide stable bias sources and high impedance loads. In the following circuit a current source provides an active load to a common-emitter stage. The gain of this stage is determined by the transconductance of the first transistor and the collector resistances of both transistors.

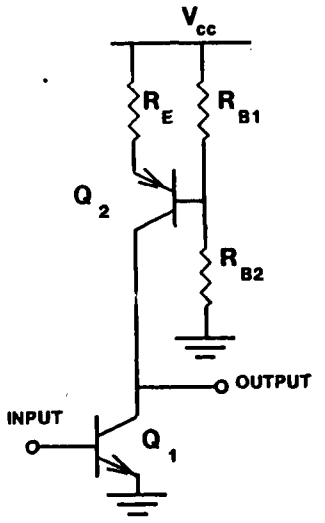


Figure 1 : Common-Emitter Amplifier with Active Load

Since the voltage on a collector is not a causal input, Q2, RE, R1 and R2 are quiescent, and QUAL determines the circuit to be a simple CE stage. The ideal current source has an infinite output impedance, and the only component which approximates this behavior is a transistor. Therefore, any quiescent collector of a transistor which is ON and whose collector voltage is deduced by an application of a KCL-heuristic at the collector node, is considered to be the output of a current source.

When QUAL constructs the primitive fragments from a mechanism graph it adds extra fragments to account for these current sources. A new fragment class SOURCE is introduced which has types VOLTAGE and CURRENT. QUAL splices the source into the mechanism graph with a JOIN of type CURRENTS:

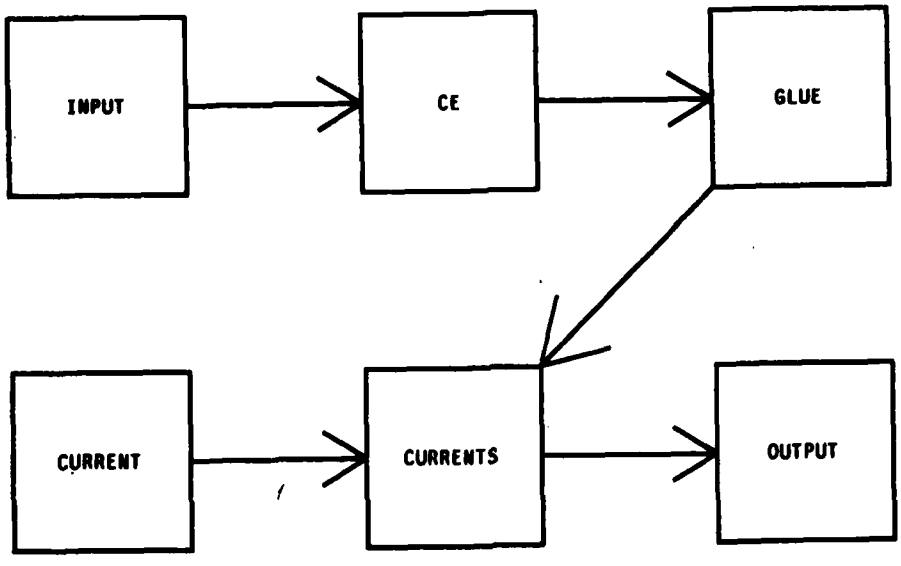


Figure 2 : Primitive Fragments for Circuit of Figure 1

The primary purpose of voltage sources is to provide references for more sophisticated current and voltage sources. QUAL considers batteries and zener-diodes which do not appear on the mechanism graph but which are connected to a stage on the mechanism graph, as basic voltage sources.

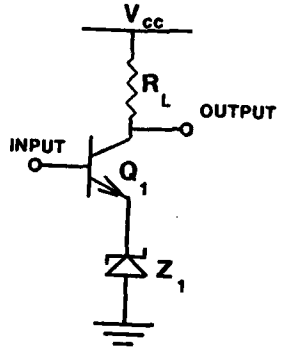


Figure 3 : Zener Bias

When QUAL constructs the primitive fragments from a mechanism graph it adds extra VOLTAGE SOURCE fragments to account for these, and modifies the stage on the mechanism graph. Typically this involves changing a CE or CB stage to a SUM differential amplifier:

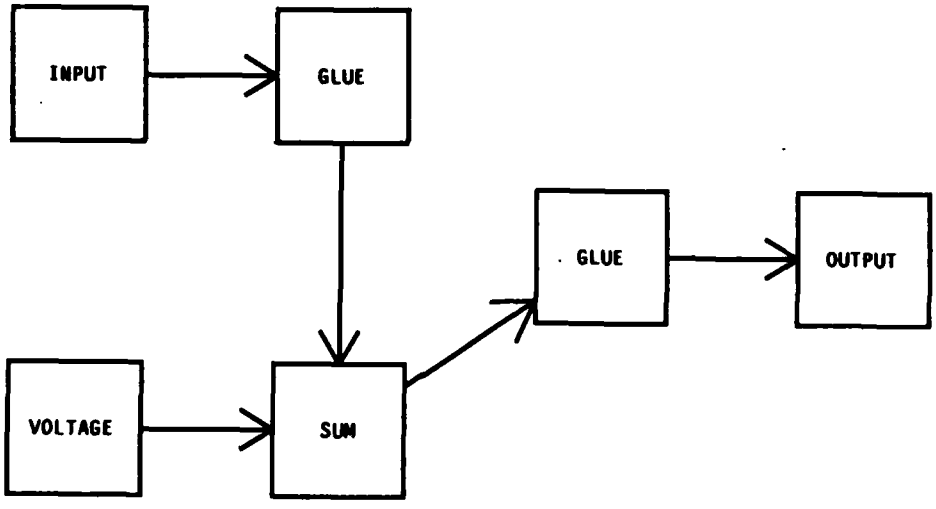


Figure 4 : Primitive Fragments for Circuit of Figure 3

QUAL does not discover any proof or rationalization that explains the functioning of sources. There are some ways in which QUAL might be extended to produce these arguments but these are not currently implemented. For example, voltage and current sources typically consist of amplifiers with no input. In the case of a current source this amplifier has a high output impedance, and in the case of a voltage source this output impedance is low. In order to identify the circuit as an amplifier its input must be identified, requiring topological searching.

It is important to recognize sources in order to determine the correct abstraction rule to apply. However, current and voltage sources can be added to any circuit *ad infinitum*, so if none of the rules in the plan library apply, the sources are removed.

8.3 *Complex Amplifiers

The abstraction rules considered so far have been rather simple and few in number, but the next three sections will introduce a number of more complex rules. Most of QUAL's rules are represented declaratively in a format much like the one used to specify circuits (see Scenario 1 of chapter 1). However, it is much easier to understand the rules when they are presented graphically. Each rule has role slots that must be filled and only fragments of a particular class (and optionally of a particular type) may fill each role. These fragments have to be connected

in a particular topology for the rule to apply. Each edge in the topological pattern is also given a role name so that coupling and glue fragments are accounted for:

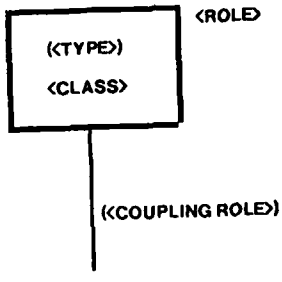


Figure 5 : Rule Pattern

The consequence of a rule is indicated by a fragment of the given type and class, and the inputs and outputs are numbered correspondingly.

The following is an example of a rule which removes voltage sources:

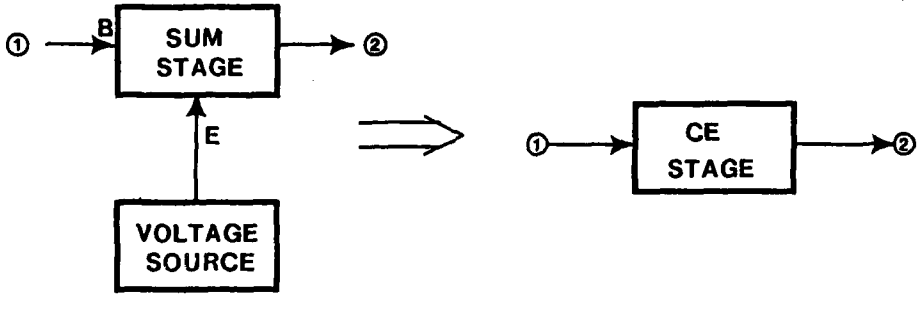


Figure 6 : Voltage Source Removal Rule

Since there are so many rules, each one is given a name. Thus far 9 rules have been

discussed:

1. FLUSH-GLUE	*Ignore GLUE and COUPLING*
2. STAGE-CASCADE	*Combine successive stages*
3. LOCAL-FEEDBACK	*Ignore local feedback*
4. UNLOOPS	*Modify sampling points*
5. FEEDBACK	*Feedback amplifiers*
6. TOP-LEVEL	*Termination rule*
7. CE-VS-BIAS	*Ignore voltage source bias*
8. CB-VS-BIAS	*Ignore voltage source bias*
9. REMOVE-CS	*Ignore current sources*

The FLUSH-GLUE rule is really an artifact of how the matcher works: since roles can only be filled by active fragments, coupling fragments are automatically ignored. The ordering of these rules is important. For example, interchanging UNLOOPS and FEEDBACK will cause faulty analyses.

Different orderings of the rules often produce different hierarchical descriptions for the same ultimate purpose. The following differential amplifier cascade illustrates this problem:

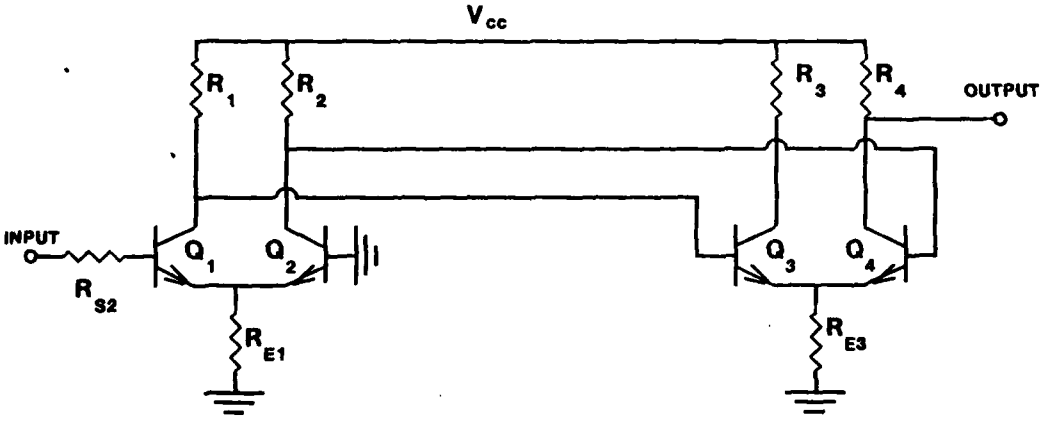


Figure 7 : Differential Amplifier Cascade

The causal analysis determines the fragments to be:

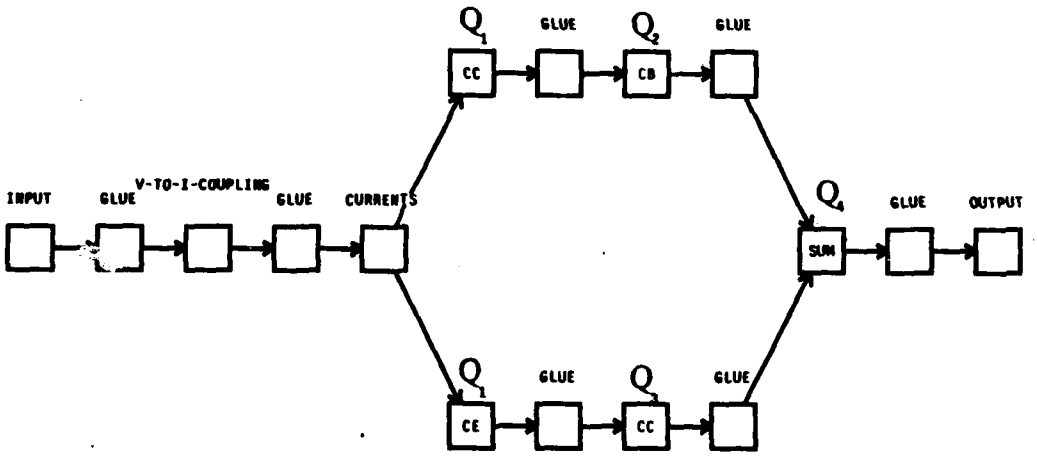


Figure 8 : Primitive Fragments for Circuit of Figure 7

The above rules will analyze this fragment graph as two parallel cascade amplifiers. (The rule for combining parallel amplifiers is missing, but that can be easily remedied.) However, an electrical engineer prefers a different parsing in which the emitter-coupled pairs are identified. To do this QUAL utilizes four new rules. An emitter-coupled pair can be recognized as a stereotypical CC stage coupled to a CB stage:

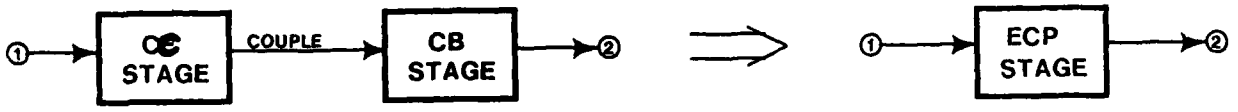


Figure 9 : Rule ECP-R-BIAS

Three more rules are required to account for multiple inputs and multiple outputs: ECP-2-1-R-BIAS, ECP-1-2-R-BIAS and CASCADE-DIFF-DIFF. The successful parse is indicated on the following fragment graph, where each of the parent fragments is labeled by the name of the rule that recognized it:

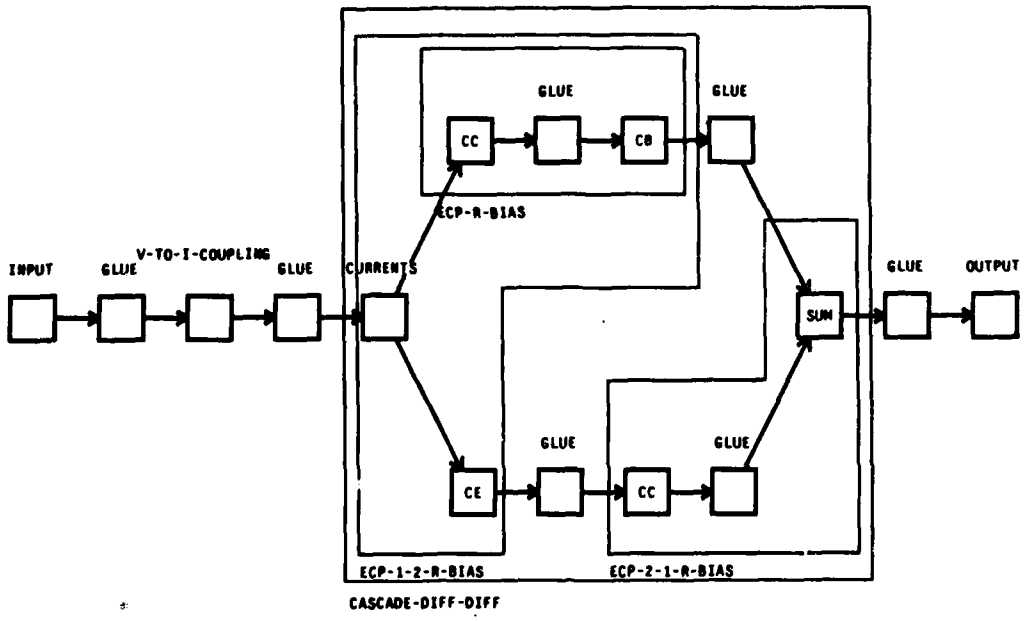


Figure 10 : Parse of Circuit of Figure 7

8.4 *Power-Supplies

QUAL considers power-supplies as voltage and current sources implemented by feedback. For power-supplies the input and output terminals are identical. A current supply tries to stabilize the output current as the voltage on its output varies, and a voltage supply tries to stabilize the voltage on its output as its output current varies. Therefore, power-supplies can be considered as amplifiers where the input quantity is the voltage or current at the output terminal, and the output quantity is the current or voltage at the same terminal. In this way, IQ analysis can deal with power-supplies. The fragment graph for the power-supply of the second scenario is:

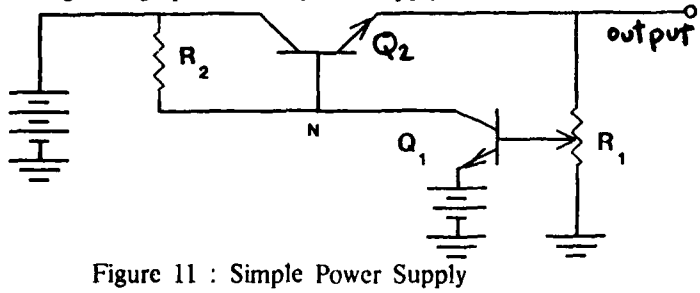


Figure 11 : Simple Power Supply

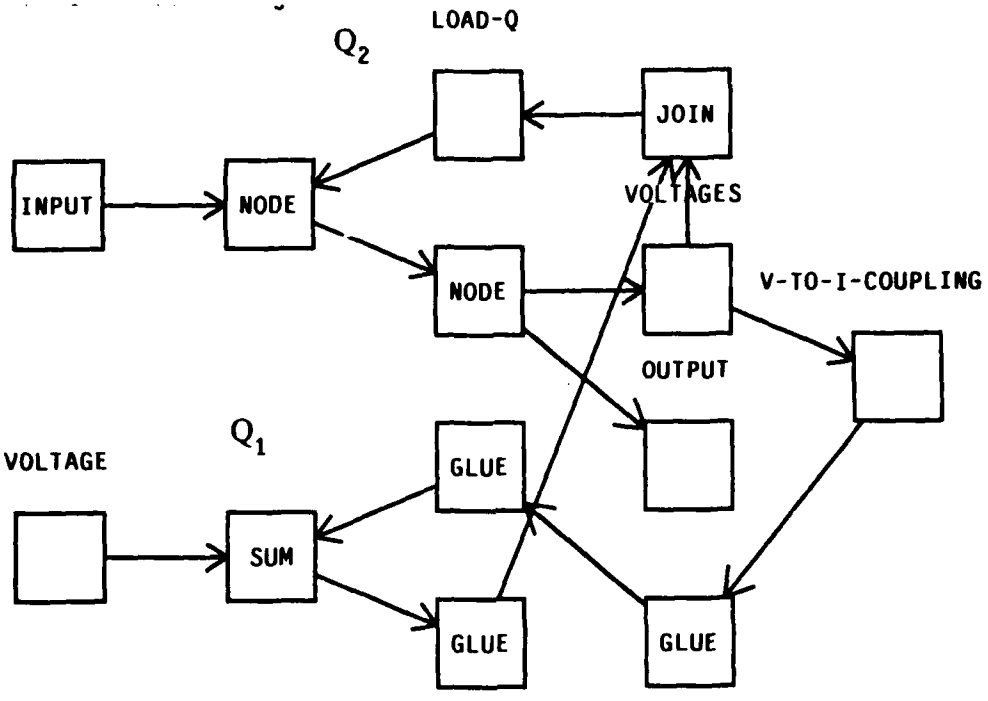


Figure 12 : Primitive Fragments for Simple Power Supply

When the rules are used to parse this fragment graph, the voltage source is ignored and Q1 and Q2 are considered as part of a cascade amplifier. This parsing is suggested by redrawing the schematic:

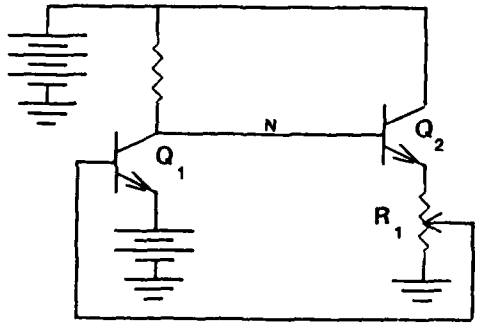
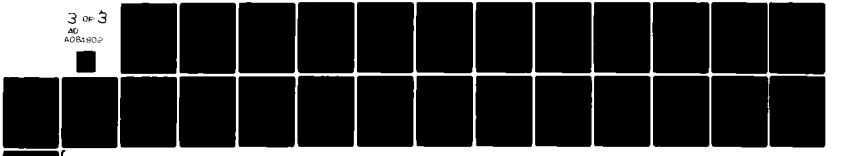


Figure 13 : Simple Power Supply as Simple Amplifier

This parsing is completely undesirable; Q1 is the comparator and Q2 the series-element of a power-supply rule:

AD-A084 802 MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTE--ETC F/G 9/5
CAUSAL AND TELEOLOGICAL REASONING IN CIRCUIT RECOGNITION.(U)
SEP 79 J DE KLEER N00014-75-C-0643
UNCLASSIFIED AI-TR-529 NL

3 OF 3
AD
A084802



END
DATE FILMED
6-80
DTIC

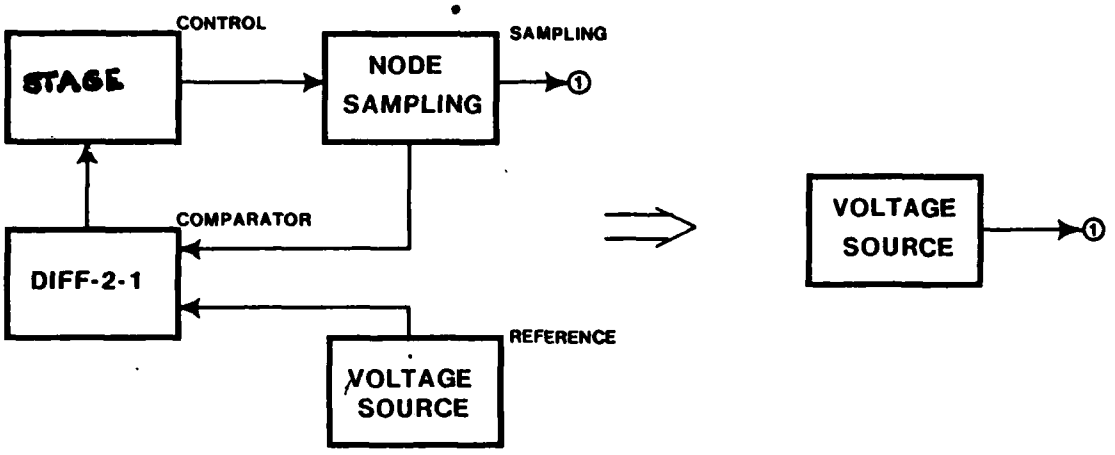


Figure 14 : Rule VOLTAGE-SIMPLE-SERIES-PASS

Before this rule can be successfully applied, two additional rules are required to deal with the peculiar output mechanism of a series-pass output section. The UNLOOPS rule has to be modified to remove the new type of local feedback, and another rule, SERIES-PASS-SIMPLE, is needed to deal with the adjacent SAMPLING and COMPARISON fragments of the node-node type feedback. QUAL's explanation for the purposes of the individual components is:

```
Q1 is functioning as SUM DIFF-2-1.
Which is COMPARATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-SIMPLE-SERIES-PASS.
Which is top-level.

Q2 is functioning as LOAD-Q COUPLING.
Which is ARTIFACT2 of CC STAGE recognized by rule UNLOOPS.
Which is CC of SERIES-PASS STAGE recognized by rule SERIES-PASS-SIMPLE.
Which is CONTROL of VOLTAGE SOURCE recognized by rule VOLTAGE-SIMPLE-SERIES-PASS.
Which is top-level.
```

```
R1A is functioning as V-TO-I-COUPLING COUPLING.
Which is COUPLING of COUPLING recognized by rule SAMPLE-COUPLE.
Which is SAMPLE-COUPLE of VOLTAGE SOURCE
            recognized by rule VOLTAGE-SIMPLE-SERIES-PASS.
Which is top-level.

R1B is functioning in I-LOAD configuration.
For a VOLTAGES SPLIT.
Which is ARTIFACT of COUPLING recognized by rule UNLOOPS.
Which is COUPLING of COUPLING recognized by rule SAMPLE-COUPLE.
Which is SAMPLE-COUPLE of VOLTAGE SOURCE
            recognized by rule VOLTAGE-SIMPLE-SERIES-PASS.
Which is top-level.

R2 is functioning in I-LOAD configuration.
For Q1 functioning as SUM DIFF-2-1.
Which is COMPARATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-SIMPLE-SERIES-PASS.
Which is top-level.

REF is functioning as VOLTAGE SOURCE.
Which is REFERENCE of VOLTAGE SOURCE
            recognized by rule VOLTAGE-SIMPLE-SERIES-PASS.
Which is top-level.
```

8.5 *Summary of the Recognition Process

In order to recognize the power-supply of Scenario 3, QUAL must utilize all of its strategies to select the correct interpretation. Many of these strategies were unnecessary in the circuits considered thus far. To summarize the recognition procedure that QUAL utilizes, each of these strategies will be summarized in the recognition process applied to this power-supply.

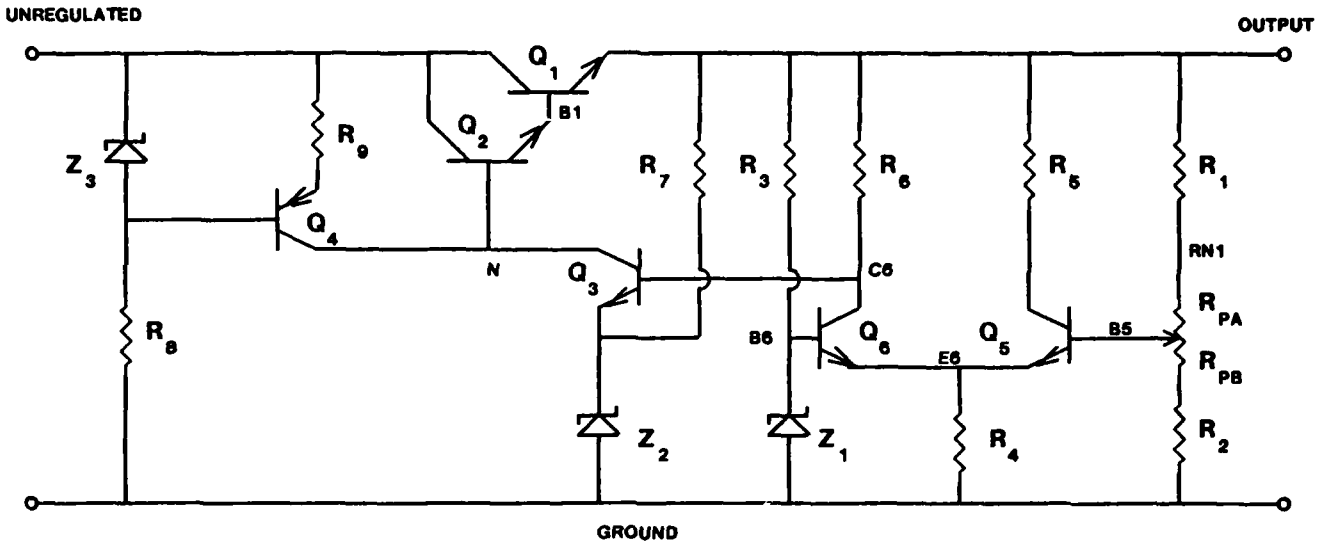


Figure 15 : Complex Power Supply

The first stage of recognition requires an envisioning which provides information about how the circuit behaves. From the envisionment, interpretations are constructed which represent the behavior of the circuit from different points of view. For this example QUAL discovers 12 interpretations, 8 of which are rejected because of unsubstantiated connection heuristics. One of these interpretations contains feedforward of a type that cannot be ruled out by other means. In this interpretation, the path R1,Q5,Q6 merges with a path through R3 and connects at the base of Q3. Since this interpretation assigns a CAUSAL purpose to R3 instead of the REFLECTION purpose of the correct interpretation, it will be chosen over the correct interpretation. The feedforward rejection rule applies since the passive coupling through R3 effectively shorts out the active stages Q6 and Q5. As a result only 3 possible interpretations remain.

QUAL has so much difficulty in selecting the correct interpretation because this circuit contains three feedback paths: a negative feedback path Q5, R5; a positive feedback path Q5, Q6, R6; and a negative feedback path Q5, Q6, Q3, Q2, Q1. The following is the mechanism graph for the correct interpretation:

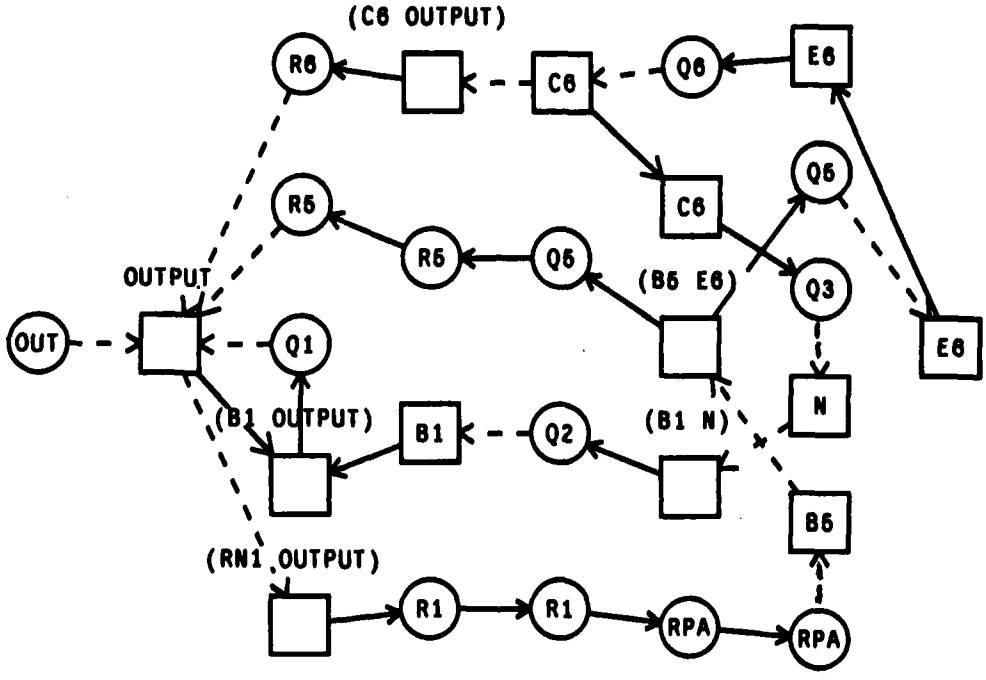


Figure 16 : Mechanism Graph for Complex Power Supply

QUAL's strategy for dealing with multiple intertwined feedback paths is to consider only the one with maximum gain. All component configurations on these ignored feedback paths are not included in the implementation teleology and are not utilzed for interpretation selection.

For each of the three remaining interpretations QUAL constructs the mechanism graph and from it, the primitive fragment graph. The implementation teleology entailed by each of these fragment graphs is compared and the one which assigns "maximum purpose" is selected.

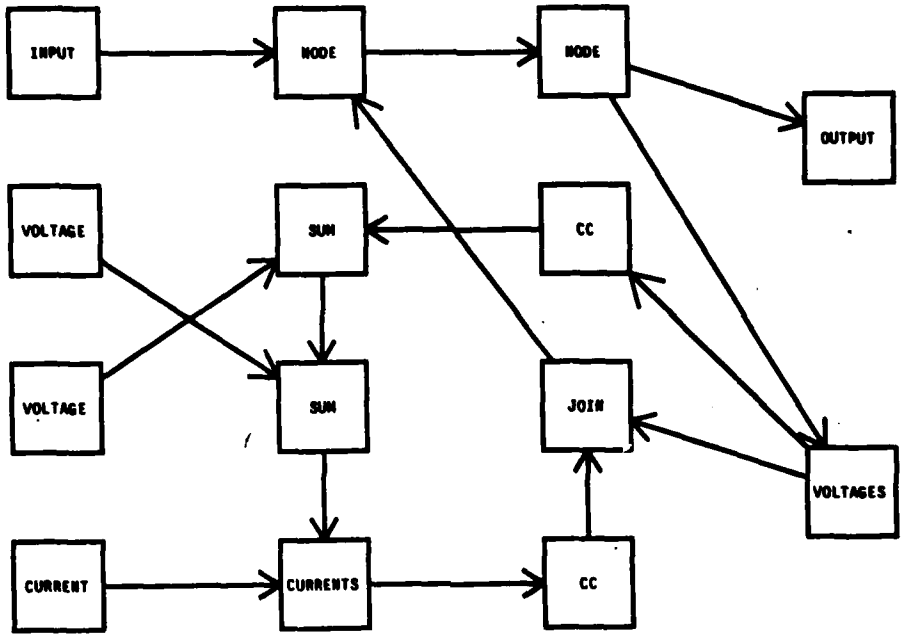


Figure 17 : Correct Fragment Graph for Complex Power Supply

QUAL requires one more cascade rule which combines a DIFF-2-1 followed by a STAGE into a DIFF-2-1. This is the CASCADE-DIFF-STAGE rule. With these rules QUAL can successfully parse the circuit as a power supply. The current source Q4 is considered to be an active load for CE stage Q3, and while this is correct, in this context it is more commonly referred to as a preregulator. Therefore, another power-supply rule is added which takes it into account. See figure 18. The resulting successful parse assigns the following purposes to each of the components.

```
Q1 is functioning as LOAD-Q COUPLING.
Which is ARTIFACT2 of CC STAGE recognized by rule UNLOOPS.
Which is CC of SERIES-PASS STAGE recognized by rule SERIES-PASS-SIMPLE.
Which is STAGE2 of CASCADE STAGE recognized by rule CASCADE.
Which is CONTROL of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.
```

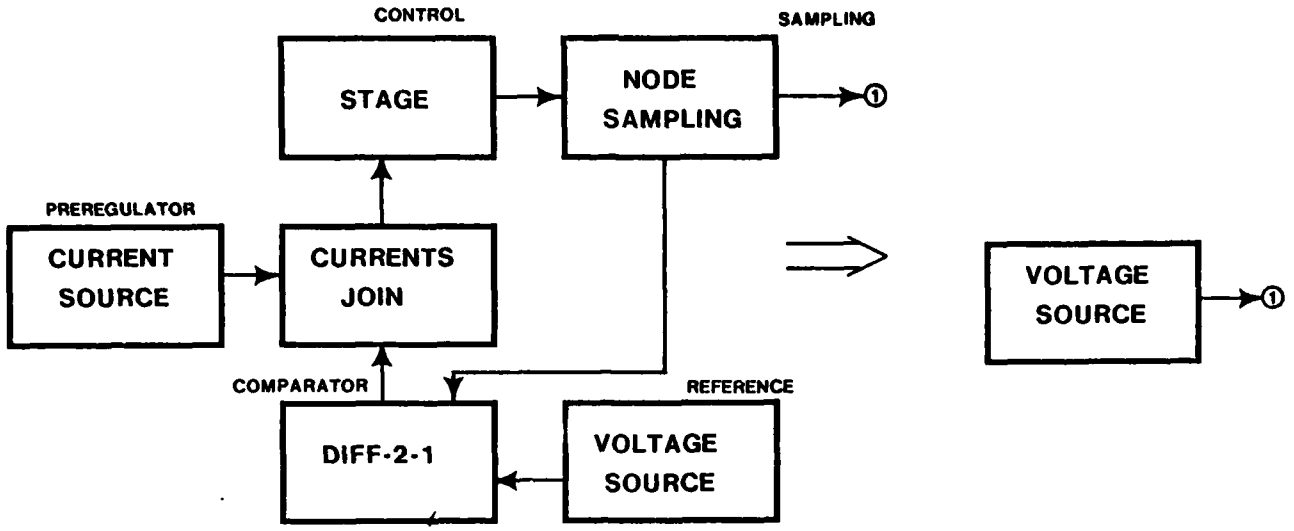


Figure 18 : Rule VOLTAGE-COMPLEX-SERIES-PASS

```
Q2 is functioning as CC STAGE.
Which is STAGE1 of CASCADE STAGE recognized by rule CASCADE.
Which is CONTROL of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

Q3 is functioning as SUM DIFF-2-1.
Which is CE of CE STAGE recognized by rule CE-CE-VS-BIAS.
Which is STAGE of CASCADE DIFF-2-1 recognized by rule CASCADE-DIFF-STAGE.
Which is COMPARATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

Q4 is functioning as CURRENT SOURCE.
Which is PRE-REGULATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.
```

```
Q5 is functioning as CC STAGE.
Which is CC of ECP DIFF-2-1 recognized by rule ECP-2-1-R-BIAS.
Which is DIFF of CASCADE DIFF-2-1 recognized by rule CASCADE-DIFF-STAGE.
Which is COMPARATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

Q6 is functioning as SUM DIFF-2-1.
Which is SUM of ECP DIFF-2-1 recognized by rule ECP-2-1-R-BIAS.
Which is DIFF of CASCADE DIFF-2-1 recognized by rule CASCADE-DIFF-STAGE.
Which is COMPARATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

R1 is functioning as V-TO-I-COUPLING COUPLING.
Which is COUPLING of COUPLING recognized by rule SAMPLE-COUPLE.
Which is SAMPLE-COUPLE of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

RPA is functioning as WIRE COUPLING.
Which is COUPLING of COUPLING recognized by rule SAMPLE-COUPLE.
Which is SAMPLE-COUPLE of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

RPB is functioning in I-LOAD configuration.
For a VOLTAGES SPLIT.
Which is ARTIFACT of COUPLING recognized by rule UNLOOPS.
Which is COUPLING of COUPLING recognized by rule SAMPLE-COUPLE.
Which is SAMPLE-COUPLE of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.
```

```
R3 is functioning in I-LOAD configuration.
For Z1 functioning as VOLTAGE SOURCE.
Which is REFERENCE of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

R4 is functioning in I-LOAD configuration.
For Q5 functioning as CC STAGE.
Which is CC of ECP DIFF-2-1 recognized by rule ECP-2-1-R-BIAS.
Which is DIFF of CASCADE DIFF-2-1 recognized by rule CASCADE-DIFF-STAGE.
Which is COMPARATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

R5 is functioning in I-LOAD configuration.
For Q5 functioning as CC STAGE.
Which is CC of ECP DIFF-2-1 recognized by rule ECP-2-1-R-BIAS.
Which is DIFF of CASCADE DIFF-2-1 recognized by rule CASCADE-DIFF-STAGE.
Which is COMPARATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

R6 is functioning in I-LOAD configuration.
For Q6 functioning as SUM DIFF-2-1.
Which is SUM of ECP DIFF-2-1 recognized by rule ECP-2-1-R-BIAS.
Which is DIFF of CASCADE DIFF-2-1 recognized by rule CASCADE-DIFF-STAGE.
Which is COMPARATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.
```

```
R7 is functioning in I-LOAD configuration.
For Z2 functioning as VOLTAGE SOURCE.
Which is BIAS of CE STAGE recognized by rule CE-CE-VS-BIAS.
Which is STAGE of CASCADE DIFF-2-1 recognized by rule CASCADE-DIFF-STAGE.
Which is COMPARATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

Z1 is functioning as VOLTAGE SOURCE.
Which is REFERENCE of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.

Z2 is functioning as VOLTAGE SOURCE.
Which is BIAS of CE STAGE recognized by rule CE-CE-VS-BIAS.
Which is STAGE of CASCADE DIFF-2-1 recognized by rule CASCADE-DIFF-STAGE.
Which is COMPARATOR of VOLTAGE SOURCE
            recognized by rule VOLTAGE-COMPLEX-SERIES-PASS.
Which is top-level.
```

The complete analysis for the complex power supply requires 3 minutes of CPU time on the KA-10, and uses 75% of the available address space of 256K. (This 192K is divided approximately: 50K basic MacLisp, 50K QUAL, and 92K temporary data structures.) Roughly 90% of this time and space is necessary to do the envisioning. The envisionment creates 188 partial environments out of 50 assumptions and constructs 513 cell values.

Chapter 9

CONCLUSION

9.1 Summary

The goal of this research was to isolate the calculus that humans tacitly employ to reason about designed systems. The preceeding chapters have presented a theory of causal and teleological reasoning, and applied it to the recognition of electronic circuits. The implementation of this theory is capable of recognizing a wide class of amplifiers and power-supplies. Success on the recognition test establishes the informative content of the two calculi. Other tests of the theory include whether it accounts for phenomenon we observe in human reasoning about circuits and its utility in applications other than recognition. Two of the phenomenon the theory explains:

1. Why some faults are dramatically harder to troubleshoot than others? (section 5.3)
2. Why it is hard for neophytes to remember explanations for circuit functioning? (section 7.3)

A variety of applications of the two calculi have been presented in some detail:

1. Analysis of novel circuits by making measurements (section 5.2).
2. Automatic troubleshooting of faulty circuits (section 5.3).
3. Standard circuit analysis and synthesis (sections 2.5, 7.10, 6.9).
4. General circuit design (sections 4.1, 7.9).
5. Education (sections 1.4, 5.3, 7.3).

This research comes out of an attempt to understand the qualitative reasoning humans employ to reason about physical systems through the study of the kinds of explanations humans give for how these systems function. Thus it is not surprising that in the specific domain chosen in this research, that the discoveries impact almost every area of human reasoning in that domain.

Although the two calculi are sufficent to succeed at the recognition task, they had to be applied across a much wider range than in human reasoning. The engineer uses a wide variety of strategies to reason about designed artifacts and it is somewhat surprising that QUAL succeeds

by employing only two of them. For example, the engineer usually places more emphasis on sophisticated teleological consderations (which QUAL is incapable of) and geometric hints, than causality. Nevertheless the methodological strategy of identifying an as simple as possible mechanism to account for the recognition has proven to be very productive.

Nevertheless, I do not claim that propagation of constraints, causal analysis, or teleological analysis have been worked out completely, but the fundamental ideas which distinguish them have been identified. The failures of the three tacit calculi identified in this research provide insight on other kinds of reasoning that humans employ. This chapter illustrates several of the types of reasoning that remain unexplained, posing suggestions for further research.

The first section presents some of QUAL's limitations having accessible solutions, while the second section discusses some types of reasoning that the calculi do not adequately explain and which do not fall into the general framework of QUAL. The distinction between mechanism and object has some consequences for design which are discussed in the third section.

9.2 Future Research I

The connection heuristics depend critically on the assumption that all signals are referenced to ground. This means that QUAL cannot distinguish between the common-mode and differential-mode behavior of an emitter-coupled pair:

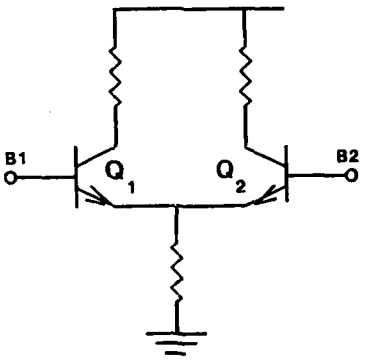


Figure 1 : Emitter-Coupled Pair

QUAL can only analyze this circuit if the the voltages at B1 and B2 are referenced to ground. If the voltage is applied between B1 and B2 (differential-mode), propagation remains incomplete since the KVL-heuristics only apply to voltages with respect to ground. Extending the KVL-

heuristic to consider any other node as a reference does not help. For example, if the voltage at B1 is increasing with respect to B2, the extended KVL-heuristic would predict that the current through Q1 is increasing and the current through Q2 is decreasing. However, the common-mode component of the signal can force both currents to increase, thereby resulting in a contradiction. This is the case when B1 and B2 are both rising, with B1 rising slightly faster than B2. If all the resistors are replaced by current sources an analogous problem arises with the KCL-heuristic.

The interaction of common-mode and differential-mode behavior is undesirable in operational amplifiers, and if these behaviors are not distinquished, the functioning of emitter-coupled pairs within these circuits can be only partially understood. Other circuits are crucially dependent on this interaction. The following analog multiplier achieves its multiplicative behavior by combining the common-mode and differential-mode behavior of the emitter-coupled pair Q1-Q2:

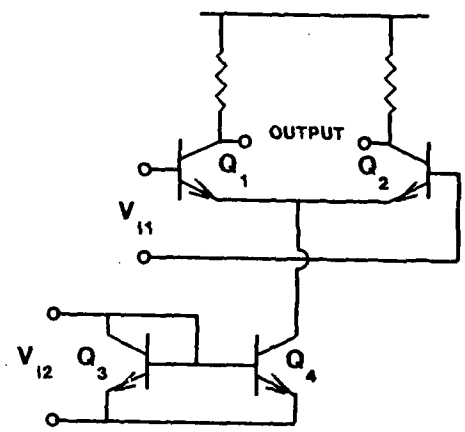


Figure 2 : Analog Multiplier

The solution to this difficulty appears to require the notion of a "relative" signal, a notion that I have not been able to crystalize. The difficulty lies in the concept of relative; if every signal is considered relative to every other, IQ analysis becomes intractable.

QUAL's plan library lacks many rules. In order to identify the missing rules the recognizer should be run on a dictionary of circuits. However, instead of using a person to add the missing rules, QUAL should be able to learn them by itself. Presented with a circuit and a description of what it does (class and type), it should parse it as far as possible incorporating the top-level pattern into its library. If QUAL is trained on a specially chosen sequence of examples, this should pose little problem and would be a convenient way of building the library. Before this is done the library rules should be extended to allow coupling components to fill roles. It is also unclear how good the interpretation selection heuristics are.

It is possible that as the library is expanded the interpretation selection heuristics may sometimes fail. Of the 50 or so examples that QUAL has been run on, it has never chosen an incorrect interpretation. In the two cases where it does fail, it cannot distinguish between two equally good interpretations. (The circuits are complementary-symmetry pairs.) In these cases the library should be utilized to parse each interpretation individually choosing the one with the more reasonable abstraction teleology.

Although QUAL can build a state transition diagram for any circuit, it does not know how to incorporate this information into the plan rules. Thus it cannot deal with logic circuits or overload protection circuitry in power supplies. This is a result of QUAL's inability to do any quiescent analysis at all, forcing it to consider many circuit states which are quiescently impossible. It appears that quiescent analysis can be done with the same basic machinery used for IQ and propagation of constraints analysis permitting the development of a new kind of mechanism description which incorporates both incremental and quiescent behavior. This description method could be general enough to be useful for describing the functioning of mechanical as well as electrical devices.

Most of the resources required to recognize a circuit are required in the IQ analysis. However, most of this analysis will eventually be determined to be contradictory or irrelevant because not enough of the higher level information is incorporated into the lower level reasoning. This problem has been partially dealt with in the current implementation by blurring the distinctions between the levels of reasoning. The criteria discussed in chapters 5 and 6 to evaluate interpretations fall into two categories: extensional criteria which are applicable only when all the possible environments

to be selected among are known, and intensional criteria which can be applied to an environment by itself. The application of the intensional criteria need not wait until the IQ analysis is finished, but can be applied to environments constructed during the IQ analysis. Only the aplication of the extensional criteria needs to wait until the interpretation generation phase. As well, QUAL only partially constructs the fragment graphs and only completes the construction of the one for the selected interpretation. These two changes to the original QUAL decreased the analysis time of the complex power supply from one hour to 3 minutes.

QUAL would benefit a great deal if some of the implementation and abstraction teleology were incorporated into the IQ analysis. The envisioning process would then more closely approximate the engineer's. A symptom of this problem is that IQ analysis does not break up the circuit into sections, and therefore ambiguities in an input section cause multiple analyses of the remaining sections of the circuit with the cost of the envisioning being exponential in the size of the circuit. If QUAL could determine where to break the circuit into sections this exponential behavior could be reduced to linear. The only effect of imposing a boundary is that KCL-heuristics on the boundary fail. If boundaries could be judiciously chosen at points which are between major stages instead of within stages, this would pose no problem.

Geometric analysis plays a major role in the engineer's recognition of circuits. The tacit calculus that is used to draw circuit schematics must be articulated. Similarly, topological analysis also plays a role. The more difficult problem is how topological and functional analysis interact in the recognition of a circuit when only part of the circuit is topologically recognized.

9.3 Further Research II

The number of types of reasoning about which we currently have little insight is staggering:

1. Reasoning about signals: impulses, square waves, ramps.
2. Reasoning in the frequency domain: filters, oscillators, phase-locked loops.
3. Reasoning about unmodelled effects: noise, temperature, physical layout.
4. Reasoning about specifications: impedances, gains.
5. Troubleshooting.
6. Reasoning about constraints.

Reasoning about constraints as opposed to reasoning with constraints avoids algebraic analysis. This type of calculus is usually not seen in beginning engineering students and becomes evident only in more experienced engineers. Reasoning about constraints is useful in understanding current sources, particularly modern integrated circuits, as well as most of the circuits considered in this thesis. This kind of reasoning involves utilizing the laws of network theory, avoiding most of the algebraic analysis. A typical circuit where reasoning about constraint is important is the input stage of a 741 operational amplifier.

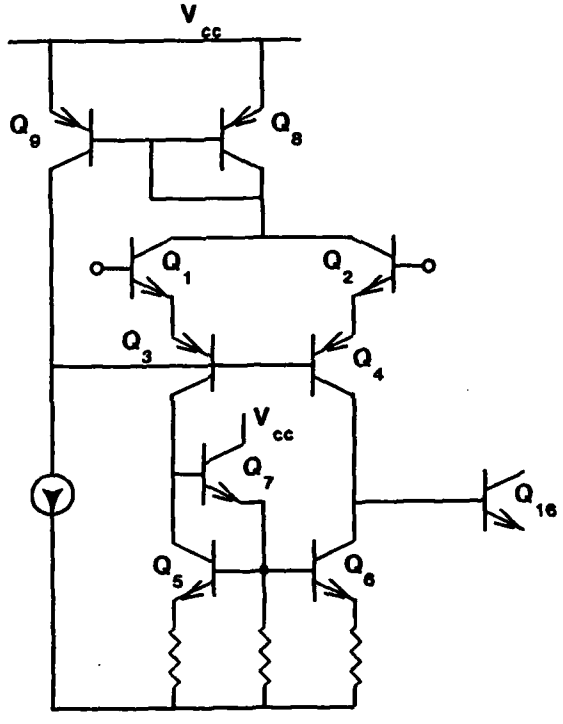


Figure 3 : 741 Input Stage

This circuit is usually explained is in terms of current constraints because one of central purposes of this stage is to select out the differential-mode from the input signal and convert it into a single-ended signal. For example, the behavior of the output node is usually explained by equal currents flowing from Q4 and Q6, thereby fixing the base current of Q16. This could also be explained, although somewhat unsatisfactorily, by the KVL-heuristics. The real difficulty for IQ analysis and the necessity for constraint reasoning is the behavior of the common-mode feedback circuit Q3-Q4-Q9-Q8.

9.4 Design

This research is part of a larger effort to understand design, a goal so far only alluded to in this thesis. Two of the suggested areas of further research discussed in the previous sections are central to the achievement of this goal: a general representation of mechanism and reasoning about constraints.

When a designer is presented with a new, desired behavior he constructs plausible abstract mechanisms by which this behavior can be achieved. From this mechanism description, he utilizes his library to construct a circuit that exhibits this mechanism. This can be a complicated process, but the point is that he constructs an intermediate representation of the mechanism, not of the circuit. Recognition is a dual of design: recognition involves identifying the abstract mechanism from a given circuit, while design requires the construction of a circuit from a given abstract mechanism. Therefore, the identification of a more general description of mechanism, incorporating both the quiescent and incremental aspects of a behavior, is a fundamental key in dealing with both design and recognition.

Most circuits can be analyzed through both causal and constraint re)ning. Causal analysis appears to be the more elementary of the two and is the one more commo. .y used by engineers. Constraint analysis is, however, required in circuit design. This is the only way SYN's intractable algebra can be avoided.

REFERENCES

[Brown 76]

Brown, A.L., "Qualitative Knowledge, Causal Reasoning, and the Localization of Failures," Artificial Intelligence Laboratory, TR-362, Cambridge: M.I.T., 1976.

[Brown *et.al.* 74]

Brown, J.S., R.R. Burton and A. G. Bell, SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (An example of AI in CAI), Final Report, B.B.N. Report 279, A.I. Report 12, Cambridge: B.B.N., 1974.

[Brown *et.al.* 76]

Brown, J.S., R. Rubinstein and R. Burton, Reactive Learning Environment for Computer Assisted Electronics Instruction, B.B.N. Report 3314, I.C.A.I. Report 1, Cambridge: B.B.N., 1976.

[Brown *et.al.* 77]

Brown, J.S., A. Collins and G. Harris, "Artificial Intelligence and Learning Strategies," B.B.N. Report 3634, I.C.A.I. Report 6, Cambridge: B.B.N., 1977. (Also in *Learning Strategies* New York: Academic. Press, 1978.)

[Bundy *et.al.* 76]

Bundy, A., Luger, G., Stone, M. and R. Welham, "MECHO: Year one," *Proceedings of the 2nd AISB Conference*, pp. 94–103, 1976.

[Stevens & Collins 78]

Stevens, A., and A. Collins, "Multiple Conceptual Models of a Complex System," B.B.N. Report 3923, Cambridge: B.B.N., 1978.

[de Kleer 75]

de Kleer, J., "Qualitative and Quantitative Knowledge in Classical Mechanics," Artificial Intelligence Laboratory, TR-352, Cambridge: M.I.T., 1975.

[de Kleer 76]

de Kleer, J., "Local Methods for Localizing Faults in Electronic Circuits," Artificial Intelligence Laboratory, AIM-394, Cambridge: M.I.T., 1976.

[de Kleer 77a]

de Kleer, J., "A Theory of Plans for Electronic Circuits," Artificial Intelligence Laboratory, WP-144, Cambridge: M.I.T., 1977.

[de Kleer 77b]

de Kleer, J., "Multiple Representations of Knowledge in a Mechanics Problem-Solver," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pp. 299–304, 1977.

[de Kleer & Sussman 78]

de Kleer, J. and G.J. Sussman, "Propagation of Constraints Applied to Circuit Synthesis," Artificial Intelligence Laboratory, AIM-485, Cambridge: M.I.T., 1978.

[de Kleer 79]

de Kleer, J., "The Origin and Resolution of Ambiguities in Causal Arguments," *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pp. 197–203, 1979.

[Doyle 77]

Doyle, J., "Truth Maintenance Systems for Problem Solving," Artificial Intelligence Laboratory, TR-419, Cambridge: M.I.T., 1977.

[Feigenbaum 77]

Feigenbaum, E.A., "THE ART OF ARTIFICIAL INTELLIGENCE: I. Themes and case studies of knowledge engineering," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pp. 1014–1029, 1977.

[Feynman 65]

Feynman, R., *The Character of Physical Law*, Cambridge: M.I.T. Press, 1965.

[Freiling 77]
Freiling, M.J., "The Use of a Hierarchical Representation in the Understanding of Mechanical Systems," Ph.D. thesis, Cambridge: M.I.T., 1977.

[Goldstein & Carr 77]
Goldstein, I. and B. Carr, "The Computer as Coach: An Atheletic Paradigm for Intellectual Education," *Proceedings of the 1977 Conference, Association for Computing Machinery*, pp. 227–233, October 1977.

[Gottlieb 75]
Gottlieb, I.M., *Regulated Power Supplies*, New York: Howard W. Sams & C., 1975.

[Gray & Searle 69]
Gray, P.E. and C.L. Searle, *ELECTRONIC PRINCIPLES Physics, Models, and Circuits*, New York: John Wiley & Sons, 1969.

[Grinberg 78]
Grinberg, M.R., "SEMI-AUTOMATIC DIGITAL DESIGNER SYSTEM — A Ph.D. Dissertation Proposal," Department of Computer Science, TR-685, Maryland: University of Maryland, 1978.

[Harris *et.al.* 66]
Harris, J.N., P.E. Gray and C.L. Searle, *Digital Transistor Circuits*, New York: John Wiley & Sons, 1966.

[Hayes 78]
Hayes, P.J., "The Naive Physics Manifesto," unpublished draft, Department of Computer Science, Essex: University of Essex, 1978.

[Larkin 77]
Larkin, J.H., "Processing Information for Effective Problem Solving," Group in Science and Mathematics Education working paper, Berkeley: University of California, 1977.

[Macsyma 77]
The Mathlab Group, *MACSYMA Reference Manual*, Laboratory for Computer Science, Cambridge: M.I.T., 1977.

[McDermott 76a]
McDermott, D.V., "Artificial Intelligence Meets Natural Stupidity," *SIGART Newsletter*, No. 57, April, 1976.

[McDermott 76b]
McDermott, D.V., "Flexibility and Efficiency in a Computer Program for Designing Circuits," Artificial Intelligence Laboratory, TR-402, Cambridge: M.I.T., 1976.

[McDermott & Larkin 78]
McDermott, J. and J.H. Larkin, "Re-representing Textbook Physics Problems," *Proceedings of the Second National Conference of the Canadian Society for Computational Studies of Intelligence*, pp. 156–164, 1978.

[Minsky 74]
Minsky, M., "Frame-Systems: A Framework for Representation of Knowledge," Artificial Intelligence Laboratory, AIM-306, Cambridge: M.I.T., 1973.

[Motorola 67]
Motorola, *Zener Diode Handbook*, Motorola, 1967.

[Rich & Shrobe 78]
Rich, C., and H.E. Shrobe, "Intial Report on a LISP Programmer's Apprentice," *IEEE Transactions on Software Engineering*, Vol. SE-4 No. 6, November 78.

[Rich *et.al.* 77]
Rich, C., Shrobe H.E., Waters R.C., Sussman G.J. and C.E. Hewitt, "Programming Viewed as an Engineering Activity," Artificial Intelligence Laboratory, AIM-459, Cambridge: M.I.T., 1977.

[Rieger & Grinberg 77]

Rieger, C., and M. Grinberg, "The Declarative Representation and Procedural Simulation of Causality in Physical Mechanisms," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pp. 250–255, 1977.

[Roylance 75]

Roylance, G.L., "Anthropomorphic Circuit Analysis," bachelor's thesis, Cambridge: M.I.T., 1975.

[Shrobe 79]

Shrobe, H.E., "Dependency Directed Reasoning for Complex Program Understanding," Artificial Intelligence Laboratory, TR-503, Cambridge: M.I.T., 1979.

[Stallman & Sussman 77]

Stallman, R.M. and G.J. Sussman, "Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis," *Artificial Intelligence 9*, pp. 135–196, 1977.

[Steele & Sussman 78]

Sussman, G.J. and G.L. Steele, "Constraints," Artificial Intelligence Laboratory, AIM-502, Cambridge: M.I.T., 1978.

[Sussman 73]

Sussman, G.J., "A Computational Model of Skill Acquisition," Artificial Intelligence Laboratory, TR-297, Cambridge: M.I.T., 1973.

[Sussman 77a]

Sussman, G.J., "Electrical Design: A Problem for Artificial Intelligence Research," Artificial Intelligence Laboratory, AIM-425, Cambridge: M.I.T., 1977.

[Sussman 77b]

Sussman, G.J., "SLICES At the Boundary between Analysis and Synthesis," Artificial Intelligence Laboratory, AIM-433, Cambridge: M.I.T., 1977.

[Sussman & Brown 74]

Sussman, G.J. and A.L. Brown, "LOCALIZATION OF FAILURES IN RADIO CIRCUITS A STUDY IN CAUSAL AND TELEOLOGICAL REASONING," Artificial Intelligence Laboratory, AIM-319, Cambridge: M.I.T., 1974.

[Sussman & Stallman 75]

Sussman, G.J. and R.M. Stallman, "Heuristic Techniques in Computer-Aided Circuit Analysis," *IEEE Transactions on Circuits and Systems*, Vol. CAS-22 No. 11, November 1975.

[Walston & Miller 63]

Walston, J.A. and J.R. Miller, *Transistor Circuit Design*, New York: McGraw-Hill, 1963.

[Waters 78]

Waters, R.C., "Automatic Analysis of the Logical Structure of Programs," Artificial Intelligence Laboratory, TR-492, Cambridge: M.I.T., 1979.

[Weinreb & Moon 79]

Weinreb, D., and D. Moon, "Lisp Machine Manual," Artificial Intelligence Laboratory, Cambridge: M.I.T., 1979.

APPENDIX 1: The IQ Models

Models are described by an association list of variable bindings followed by a specification of the relationships which hold among these variables.

```
(((v (voltage e))                    ;association for emitter potential
  (ic (current c)))                  ;association for collector current
 (increasing v implies increasing ic))    ;rule prototype
```

`Voltage` and `current` refer to the appropriate cell of the specific transistor.

* * *

```
(((v (voltage #1 #2))
  (i (current #1)))
 (↕v <=> ↕i)))
```

Model 1

The `#1` and `#2` which appear in the association list refer to the two terminals of the resistor.

* * *

```
(((v (voltage #1 #2))
  (i (current #1)))
 (choice (on  (0 => ↕v))
         (off (0 => ↕i))))
```

Model 2

The `choice` construct specifies the rule prototypes that apply for each of the regions of operation of the device.

* * *

```
(((v (voltage b e))
  (ib (current b))
  (ic (current c))
  (ie (current e)))
 (choice (on (0 => ↕v) (↕ib => ↕ic) (↕ib -=> ↕ie))
         (off (0 => ↕ib) (0 => ↕ic) (0 => ↕ie))
         (sat (0 => ↕v) (0 => ↕ic))))
```

Model 3

Note that => and <=> always refer to derivatives. The "-=>" operator behaves like => except that it inverts the sign of the assigned quantity.

* * *

```
(choice (on (↕v => ↕ib) (↕v => ↕ic) (↕v -=> ↕ie)))
```

Model 4

* * *

```
(choice (on (0 => ↕v) (↕ib => ↕ic) (↕ib -=> ↕ie))
```

Model 5

* * *

```
(((v (voltage #1 #2))
  (v1 (voltage-to-reference #1))
  (v2 (voltage-to-reference #2))
  (i (current #1)))
 (choice (on ()
             (↕v (=> ↕i))
             (↕v1 (C=> ↕v2) (C=> ↕i) (C=> ↕v))
             (↕v2 (C=>↕v1) (C-=> ↕i) (C-=> ↕v)))
         (off ((0 => ↕i)))))
```

Model 6

The `choice` construct has been slightly modified. The first expression of a choice lists the state-values, and the remaining rule prototypes are grouped together according to input variable.

```
(((v (voltage b e))
  (ve (voltage-to-reference e))
  (vb (voltage-to-reference b))
  (ib (current b))
  (ic (current c))
  (ie (current e)))
 (choice (on ()
             (↕v (=> ↕ic) (-=> ↕ie) (=> ↕ib))
             (↕vb (C=> ↕ve) (C=> ↕v) (C=> ↕ib) (C-=> ↕ie) (C=> ↕ic))
             (↕ve (C=> ↕vb) (C-=> ↕v) (C-=> ↕ib) (C=> ↕ie) (C-=> ↕ic)))
         (off ((0 => ↕ib) (0 => ↕ic) (0 => ↕ie)))
         (sat ((0 => ↕ic)))))
```

Model 7

```
(((v (voltage b e))
  (ib (current b))
  (ic (current c))
  (ie (current e)))
 (choice (on ()
             (↕v (=> ↕ic) (-=> ↕ie) (=> ↕ib)))
         (off ((0 => ↕ib) (0 => ↕ic) (0 => ↕ie)))
         (sat ((0 => ↕ic)))))
```

Model 8

* * *

```
(((v (voltage #1 #2))
  (i (current #1)))
 (choice (on ()
             (↕v (=> ↕i)))
         (off ((0 => ↕i)))))
```

Model 9

* * *

```
((vbe (voltage b e))
 (vce (voltage c e))
 (ib (current b))
 (ic (current c))
 (ie (current e)))
(choice (on ((= ib +) (= ic +) (= ie -) (= vbe +) (= vce +)))
        (off ((= ib 0) (= ic 0) (= ie 0)))
        (sat ((= ib +) (= ic +) (= ie -) (= vbe +) (= vce +))))
```

Model 10

* * *

```
(((v (voltage b e))
  (ib (current b))
  (ic (current c))
  (ie (current e)))
 (choice (on ()
             (↕v (=> ↕ic) (-=> ↕ie) (=> ↕ib)
                 (if ⇓ (S-> sat))
                 (if ⇑ (S-> off))))
         (off ((0 => ↕ib) (0 => ↕ic) (0 => ↕ie))
              (↕v (if ⇓ (S-> on))))
         (sat ((0 => ↕ic))
              (↕v (if ⇑ (S-> on))))))
```

Model 11

* * *

```
(((v (voltage #1 #2))
  (i (current #1)))
 (choice (on ()
             (↕v (=> ↕i) (if ⇑ (S-> off))))
         (off ((0 => ↕i))
              (↕v (if ⇓ (S-> on))))))
```

Model 12

* * *

```
  ((v (voltage #1 #2))
   (v1 (voltage-to-reference #1))
   (v2 (voltage-to-reference #2))
   (i (current #1))
   (choice (strt nil
                 (v1 (C=> v2)
                     (if ⇓ (S-> chg+))
                     (if ⇑ (S-> chg-)))
                 (v2 (C=> v1)
                     (if ⇓ (S-> chg-))
                     (if ⇑ (S-> chg+)))
                 (i (0 => v)
                 (if ⇓ (S-> chg+))
                 (if ⇑ (S-> chg-))))
           (chg+ ((⇑ => i) (S-> strt)))
           (chg- ((⇓ => i) (S-> strt)))))
```

Model 13